НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

«До захисту допущено»

Завідувач кафедри
_____ Віталій РОМАНКЕВИЧ

“___” червня 2020 р.

Дипломний проект

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Комп’ютерні системи та компоненти»

зі спеціальності **123 «Комп'ютерна інженерія»**

на тему: Система інформаційної безпеки смарт-контрактів на основі технології блокчейн

Виконала: студентка IV курсу, групи КВ-61
(шифр групи)

Сміюн Інна Володимирівна
(прізвище, ім’я, по батькові) (підпис)

Керівник доц. каф. СПіСКС к.т.н., доцент Орлова М.М.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М.
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2020 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та компоненти»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)
«__»_____20___р.

ЗАВДАННЯ
на дипломний проєкт студента
Сміюн Інна Володимирівна

1. Тема проєкту «Система інформаційної безпеки смарт-контрактів на основі технології блокчейн»,
керівник проєкту доц. каф. СПіСКС к.т.н., доцент Орлова М.М.,
затверджені наказом по університету від «__»_____20__р. №_____

2. Термін подання студентом проєкту 22.05.2020

3. Вихідні дані до проєкту Назва. Система інформаційної безпеки смарт-контрактів на основі технології блокчейн.

4. Зміст пояснювальної записки

1. Аналіз існуючих рішень та обґрунтування теми дипломного проєкту
2. Технологія Blockchain та смарт-контракти у мережі Ethereum
3. Аналіз розробки та роботи програмного продукту

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) презентація; структурна схема організації блоків у технології блокчейн; схема алгоритму роботи смарт-контракту; структурна схема процесу розробки та взаємодії смарт-контракту у мережі Ethereum

blockchain; структурна схема принципу взаємодії основних модулів смарт-контракту.

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
нормоконтроль	Ярослав КЛЯТЧЕНКО доц. каф. СПСКС, к.т.н.		

7. Дата видачі завдання 17.09.2019.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	05.02.2020	
2.	Розроблення та узгодження технічного завдання	14.02.2020	
3.	Аналіз існуючих рішень	01.03.2020	
4.	Підготовка матеріалів першого розділу дипломного проєкту	22.03.2020	
5.	Підготовка матеріалів другого розділу дипломного проєкту	07.04.2020	
6.	Підготовка матеріалів третього розділу дипломного проєкту	22.04.2020	
7.	Підготовка графічної частини дипломного проєкту	02.05.2020	
8.	Оформлення документації дипломного проєкту	09.05.2020	
9.	Попередній огляд матеріалів диплому на кафедрі	20.05.2020	

Студент _____ Інна СМІЮН
(підпис)

Керівник проєкту _____ Марія ОРЛОВА
(підпис)

_____

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (64 с., 32 рис., список використаної літератури з 29 найменувань, 3 додатки).

Об’єктом дослідження являється технологія Blockchain для забезпечення інформаційної безпеки, на основі якої в роботі розроблено смарт-контракт на основі приватного Blockchain Ethereum.

Мета проєкту – розробка смарт-контракту, який забезпечує надійну передачу інформації та є захищеним від несанкціонованого доступу на основі вивчення та опрацювання теоретичного підґрунтя технології Blockchain, аналізу існуючих проблем інформаційної безпеки та способів їх вирішення.

У ході роботи над дипломним проєктом було:

- досліджено проблеми та загрози інформаційної безпеки, проаналізовано існуючі методи їх вирішення;
- вивчено та проаналізовано особливості технології блокчейн, способи її будови, загальну роботу системи, а також оцінено її переваги та недоліки;
- досліджено різновид угод у формі смарт-контрактів, розглянуто існуючі платформи для їх реалізації, а також принцип їх роботи;
- розроблено структуру приватної мережі Blockchain Ethereum;
- розроблено смарт-контракт та опубліковано його у власній мережі.

Ключові слова: інформаційна безпека, кіберзагрози, технологія Blockchain, смарт-контракти, консенсус, Ethereum, Web3.js, Remix, Geth.

ANNOTATION

Qualification work includes an explanatory note (64 pages, 32 figures, list of references from 29 items, 3 appendices).

The object of research is Blockchain technology for information security, on the basis of which development is a smart contract based on a private Blockchain Ethereum.

The purpose of the project is to study and develop the theoretical basis of Blockchain technology, analysis of existing information security problems and their evaluation, development of a smart contract that will ensure reliable transmission of information and will be protected from unauthorized access.

During the development of the diploma project were:

— researched problems and threats of information security are investigated, and the existing methods of their solution;

— studied Blockchain technology, features of its structure, the general work of system, and also analyzes the advantages and disadvantages;

— the variety of agreements in the form of smart contracts is investigated, the existing platforms for their implementation are expanded, as well as the principle of their work;

— created a private network Blockchain Ethereum;

— eveloped a smart contract and published in its own network;

Keywords: information security, cyber threats, Blockchain technology, smart contracts, consensus, Ethereum, Web3.js, Remix, Geth.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045470.002 ТЗ	Система інформаційної безпеки смарт-контрактів на основі технології блокчейн Технічне завдання	4		
	А4	ІАЛЦ.045470.003 ТП	Система інформаційної безпеки смарт-контрактів на основі технології блокчейн Відомості технічного проєкту	2		
	А4	ІАЛЦ.045470.004 ПЗ	Система інформаційної безпеки смарт-контрактів на основі технології блокчейн Пояснювальна записка	64		

Змін.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.045470.001 ОА		
Розробив		Сміюн І.В.			Система інформаційної безпеки смарт-контрактів на основі технології блокчейн **Опис альбому**	Літ.	Аркуш 1 / Аркушів 2
Перевірив		Орлова М.М					
Консульт.						КПІ ім. Ігоря Сікорського, ФПМ КВ-61	
Н. контроль		Клятченко Я.М.					
Зав. каф.		Романкевич В.О.					

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045470.005 Д1	Організація блоків	1		
			у технології блокчейн.			
			Схема структурна.			
	А4	ІАЛЦ.045470.006 Д1	Алгоритм роботи	1		
			смарт-контракту.			
			Схема алгоритму.			
	А4	ІАЛЦ.045470.007 Д1	Процес розробки та	1		
			взаємодії смарт- контракту			
			у мережі			
			Ethereum blockchain.			
			Схема структурна.			
	А4	ІАЛЦ.045470.008 Д1	Принцип взаємодії	1		
			основних модулів			
			смарт-контракту.			
			Схема структурна.			
		Диск CD-ROM	Текст пояснювальної	1		
			записки.			
			Графічні матеріали.			

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Система інформаційної безпеки смарт-контрактів на основі технології блокчейн».

Галузь застосування: фінансова сфера.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проєктування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є дослідження проблем інформаційної безпеки, вивчення технології Blockchain та розробка смарт-контракту для забезпечення інформаційної безпеки.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- додавання нового учасника смарт-контракту;
- можливість управління коштами;

- видача нових монет власником контракту;
- переказ коштів між учасниками контракту та третіми особами;
- визначення прав інших учасників.

5.2. Вимоги до апаратного забезпечення

- Процесор: Intel Core i3-4005U;
- Оперативна пам’ять: 8 Гб;
- Наявність доступу до мережі Internet (GPRS, EDGE, 3G, 4G).

5.3. Вимоги до програмного та апаратного забезпечення користувача

- Операційна система Windows, Linux, OS X;
- Internet-браузер (Google Chrome, Safari, Opera, Internet Explorer або інший);
- Середовище Remix;
- Платформа Geth.

6. ЕТАПИ РОЗРОБКИ

з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	05.02.2020	
2.	Розроблення та узгодження технічного завдання	14.03.2020	
3.	Аналіз існуючих рішень	01.03.2020	
4.	Підготовка матеріалів першого розділу дипломного проєкту	22.03.2020	
5.	Підготовка матеріалів другого розділу дипломного проєкту	07.04.2020	
6.	Підготовка матеріалів третього розділу дипломного проєкту	22.04.2020	
7.	Підготовка графічної частини дипломного проєкту	2.05.2020	
8.	Оформлення документації дипломного проєкту	09.05.2020	
9.	Попередній огляд матеріалів диплому на кафедрі	20.05.2020	

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045470.004 ПЗ	Система інформаційної	64		
			безпеки смарт-контрактів			
			на основі			
			технології блокчейн			
			Пояснювальна записка			
	А4	ІАЛЦ.045470.005 Д1	Організація блоків	1		
			у технології блокчейн.			
			Схема структурна.			
	А4	ІАЛЦ.045470.006 Д1	Алгоритм роботи	1		
			смарт-контракту.			
			Схема алгоритму.			
	А4	ІАЛЦ.045470.007 Д1	Процес розробки та	1		
			взаємодії смарт- контракту			
			у мережі			
			Ethereum blockchain.			
			Схема структурна.			

Змін.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.045470.003 ТП		
Розробив		Сміюн І.В.			Система інформаційної безпеки смарт-контрактів на основі технології блокчейн **Відомість технічного проєкту**	Літ.	Аркуш 1 / Аркушів 1
Перевірив		Орлова М.М					
Консульт.						КПІ ім. Ігоря Сікорського, ФПМ КВ-61	
Н. контроль		Клятченко Я.М.					
Зав. каф.		Романкевич В.О.					

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045470.008 Д1	Принцип взаємодії	1		
			основних модулів			
			смарт-контракту.			
			Схема структурна.			
		Диск CD-ROM	Текст пояснювальної	1		
			записки.			
			Графічні матеріали.			

Змін.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.045470.003 ТП	Арк. 2

ЗМІСТ

Перелік скорочень, умовних позначень, термінів

ІБ	– інформаційна безпека.
Оракули	– група алгоритмів, що переводять інформацію, що знаходиться за межами мережі, у зрозумілий блокчейну формат
Смарт-контракт	– програмний код, що зберігається у децентралізованій, розподіленій мережі Blockchain, а також визначає умови, з якими згодні всі сторони, що використовують контракт.
ABI	– Application Binary Interface.
Blockchain	– вибудувана за певними правилами безперервна послідовна ланцюжків блоків, що містять інформацію у децентралізованій базі даних.
DDoS	– Distributed denial-of-service attack
EVM	– Ethereum Virtual Machine.
Geth	– написаний на мові програмування Go клієнт, який виконує функцію ноди в мережі Ethereum.
IoT	– Internet of Things.
P2P	– Peer-to-peer. Комп’ютерна мережа, заснована на рівності учасників, де кожен вузол виступає клієнтом та виконує функції сервера.
PoS	– Proof-of-stake.
PoW	– Proof-of-work.
URL	– Uniform Resource Locator.

ВСТУП

Стрімкий розвиток нових технологій протягом останнього десятиліття призвів до того, що Інтернет став невід'ємною частиною буденного життя для більшості людей. З появою всесвітньої павутини доступ до інформації став відкритим та вільним. Це дозволило скоротити час пошуку інформації та спростити процес обміну, проте, в свою чергу викликало ряд проблем, що потребують сучасних рішень.

Останнім часом дедалі більше галузей не встигають за стрімким і безперервним процесом розвитку, і вдосконаленням технологічних інновацій. Системи розвинуті нерівномірно, працюють згідно застарілих технологій та правил, тому іноді бувають досить повільними та ненадійними. Вони централізовані, а тому постійно піддаються витокам інформації та хакерським атакам. Вирішити всі перераховані вище проблеми і недоліки сучасних систем може технологія Blockchain.

З розвитком електронних систем неодноразово виникали ідеї створити електронний аналог готівки для віддаленої оплати, і саме тому з'явилась перша криптовалюта Bitcoin, а разом з нею і технологія Blockchain.

Криптовалюта – це фінансова одиниця, яка існує тільки у віртуальному просторі, а облік внутрішніх розрахункових одиниць забезпечує децентралізована платіжна система. Для забезпечення фінансових операцій використовуються криптографічні функції [1].

Кожна така валюта побудована за технологією Blockchain.

Термін Blockchain стає зрозумілим зі своєї назви, не важко здогадатись, що це – ланцюг блоків. Для створення нового блоку необхідно спочатку послідовно зчитати всю інформацію про старі блоки. Основа даної технології у розподіленому зберіганні інформації. Головними характеристиками являються: прозорість, анонімність, децентралізація та захищеність даних [1].

Сучасне життя потребує вирішення багатьох питань за допомогою переговорів, підпису документів, складання угод. З появою платформи Ethereum почали розвиватись смарт-контракти. Технологія Ethereum дає можливість реєстрації будь-яких угод з будь-якими активами на основі розподіленої бази контрактів типу блокчейн, не вдаючись до традиційних юридичних процедур [2].

Смарт-контракт – це угода між двома особами у вигляді комп'ютерного коду. Контракт працює на основі блокчейну, і саме тому, всі дані зберігаються в загальнодоступній базі даних та не підлягають зміненню.

Проблема інформаційної безпеки (ІБ) є надзвичайно актуальною на даному етапі розвитку інформаційних технологій. У міру розвитку сучасних інновацій, виникає дедалі більше факторів, які несуть загрози системам безпеки, а конфіденційна інформація стала вразливою, як ніколи. Інформаційна безпека забезпечує стан захищеності даних, при якому інформація являється конфіденційною, доступною та цілісною та є захищеною від несанкціонованого доступу, використання чи руйнування [3].

Блокчейн стає безцінним інструментом, що ідеально підходить для забезпечення безпеки. Технологія дозволяє передавати інформацію безпечним способом, а також використовується для запобігання маніпуляцій з даними та їх фальсифікації. Оскільки – природа блоків незмінна, використовуючи послідовне хешування разом з криптографією у децентралізованій структурі, можна побудувати систему, маніпулювання якою буде практично неможливим.

Провівши аналіз актуальності проблеми, було вирішено дослідити та проаналізувати забезпечення інформаційної безпеки на основі технології блокчейн, з використанням смарт-контрактів та розробити смарт-контракт на основі приватного блокчейну Ethereum, що буде повністю захищеним та безпечним.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ

1.1. Аналіз актуальності задачі

З появою та розвитком інформаційних технологій інформаційна безпека стала відігравати провідну роль у суспільстві. Нині в мережі Інтернет не лише користуються електронною поштою, пошуковими системами, але і ведеться бізнес, займаються торгівлею, здійснюються покупки, ведуться приватні переговори, обмінюються секретною інформацією та багато іншого. Глобальна інформатизація, стрімкий розвиток техніки та цифрових технологій вимагають термінового підвищення якості забезпечення інформації. Захист інформації тісно пов’язаний з галузями міжнародного права, організації управління, розвитком технічних засобів та технологій.

Будь-якому інформаційному ресурсу, який містить у собі конфіденційну інформацію потрібна надійна система безпеки. У міру того, наскільки спрощується доступ до інформації, загрози безпеки набувають все більш масштабного характеру. Інформаційна безпека повинна передбачити у комплексі всі потенційні та реальні загрози, які можуть нести шкоду інтересам особистості, суспільства чи держави.

Лише 3% вебдодатків достатньо надійні, щоб протистояти хакерам, 97% вебсайтів мають дефекти у захисті, у результаті чого дані та системи можуть бути зламані. Серед виявлених дефектів, майже 40% сайтів дозволяють хакерам отримувати повний контроль і доступ до інформації. Близько 23% дефектів могли привести до порушенні конфіденційності, а 21% виявлених помилок давали можливість «викрадати» товари з інтернет-магазинів. 5% дозволяли хакерам змінювати інформацію, а ще 5% - перехоплювати транзакції. 2% помилок наскільки серйозні, що зловмисники можуть спокійно видалити вебсайти.

У більш розвинутих країнах установи витрачають на забезпечення інформаційної безпеки від 5 до 7% бюджету, відведеного на інформаційні технології. У нашій державі компанії витрачають на ці цілі, на жаль, лише близько 1-2% [4].

Проблема забезпечення безпеки є однією з найактуальніших у сучасному світі. Зростає кількість кіберзагроз, у тому числі й пов'язаних з крадіжкою ідентифікаційних даних. За інформацією аналітичного агентства Cybersecurity Ventures, щорічний збиток від кіберзлочинів досягне 6 трильйонів доларів до 2021 року. У 2015 році збиток становив 3 трильйони. Зростає і кількість коштів, вкладених у кібербезпеку - до 2021 року сума перевищить 1 трильйон доларів [4].

Хакерство перетворилось на серйозну та поширену проблему віртуального простору, організації зі всього світу намагаються усунути недоліки у своїх мережах та забезпечити їхню цілісність проти майбутніх кібератак.

Ведеться багато дискусій на рахунок визначення способів забезпечення цілісності інформаційної безпеки, дане питання перебуває на стадії розробки. Основним завданням є забезпечення інформаційного суверенітету найбезпечнішим методом. Технологія Blockchain уже набула широкого використання у криптографії, оскільки, вона дозволяє передавати інформацію дуже безпечним та надійним методом, а також використовується для запобігання маніпуляцій з даними. Інформаційна безпека стає все ближче до адаптації блокчейн, що дозволить забезпечити безпеку цифрової «особистості» користувача, передачу інформації та захист різних галузей.

1.2. Аналіз основних загроз інформаційної безпеки та об'єктів зони ризику

Загрозами інформаційної безпеки вважають сукупність факторів та умов, які створюють небезпеку порушення інформаційної безпеки, тобто, це

дії чи вплив, процеси чи явища, які можуть прямо чи опосередковано призвести до нанесення шкоди інтересам осіб, суспільства чи держави [5].

Розрізняють різні класифікації, в залежності від загроз, що шкодять ІБ. Це загрози цілісності, конфіденційності, доступності, які можуть бути як внутрішніми, так і зовнішніми, а також загальними, локальними чи приватними.

Також розрізняють ряд ресурсів інформаційної безпеки, які знаходяться у зоні ризику. Серед основних об'єктів виділяють наступні:

— Макрооб'єкти. Це будівлі, у яких містяться ресурси інформаційної системи, а також системи кондиціонування та інше підтримуюче обладнання. Макрооб'єкти зазвичай знаходяться під ризиком форс-мажорних обставин, у основному різних стихійних явищ.
— Устаткування. Це апаратне забезпечення інформаційної системи, яке може вийти з ладу. Сюди не відносять об'єкти, що розташовані поза межами макрооб'єктів організації.
— Програмне забезпечення. Усі програмні продукти та документація, які можуть бути втрачені у разі руйнування інформаційної системи. У цю категорію входять як комерційні програми, так і програми власної розробки.
— Носії інформації.
— Дані. Інформація, яка є необхідною для функціонування організації, а також інтелектуальна власність, персональні дані працівників.

1.3. Аналіз існуючих методів забезпечення інформаційної безпеки

Для того, щоб протистояти перерахованим у попередньому розділі загрозам, сучасні інформаційні системи включають у себе підсистеми безпеки. Політика безпеки, в залежності від цілей і умов функціонування системи може

визначати права доступу суб'єктів до ресурсів, регламентувати порядок дій користувачів у системі, захищати мережеві комунікації, формулювати способи відновлення системи після випадкових збоїв і тому подібне. Для забезпечення безпеки існують правові, організаційно-адміністративні та інженерно технічні заходи захисту інформації.

Правове забезпечення безпеки інформації – це сукупність законодавчих актів, нормативно-правових документів, положень та інструкцій, вимоги яких є обов'язковими у системі захисту інформації [6]. У нашій країні правові основи забезпечення безпеки комп'ютерних систем складають: Конституція України, Закони України, постанови Кабінету Міністрів України, кодекси та інші нормативно-правові акти.

Організаційно-адміністративне забезпечення безпеки інформації – це регламентація виробничої діяльності та взаємовідносин виконавців на нормативно-правовій основі таким чином, щоб розголошення, витік і несанкціонований доступ до інформації ставав неможливим за рахунок проведення організаційний заходів [6]. До заходів цього класу можна віднести: підбір і навчання персоналу, визначення посадових інструкцій працівників, організацію пропускного режиму, охорону приміщень, визначення порядку зберігання інформації, резервування, знищення конфіденційної інформації і тому подібне.

Інженерно-технічні заходи являють собою сукупність спеціальних органів, технічних заходів та засобів, що спільно функціонують для виконання певного завдання щодо захисту інформації. До інженерних засобів відносять екранування приміщень, організація сигналізації, охорона приміщень з ПК.

Технічні засоби захисту включають у себе як апаратні, так і програмні, криптографічні засоби захисту, які ускладнюють можливість атаки, допомагають виявити факт її виникнення, або ж позбутись від наслідків атаки. Вони виконують такі основні функції [7]:

— автентифікація партнерів по взаємодії, що дозволяє переконатися в достовірності партнера при встановлення з'єднання;
— автентифікація джерела інформації, що дозволяє переконатися в достовірності джерела повідомлення;
— управління доступом, що забезпечує захист від несанкціонованого використання ресурсів;
— конфіденційність даних, яка забезпечує захист від несанкціонованого отримання інформації;
— цілісність даних, що дозволяє виявити, а в деяких випадках і запобігти зміні інформації при її зберіганні та передачі;

1.4. Застосування технології Blockchain у цілях забезпечення інформаційної безпеки

Основними характеристиками технології є децентралізованість, незмінність та прозорість. Саме ці риси роблять блокчейн відмінним рішенням для питань інформаційної безпеки.

Децентралізація блокчейну значно знижує ймовірність фальсифікації баз даних. Спосіб, за допомогою якого хакерам вдається отримати інформацію, полягає у атаці того місця, де кластеризовані всі дані – мейнфрейму. У блокчейні це практично неможливо, оскільки вся інформація зберігається у розподіленій базі даних, а тому не існує єдиного місця, яке необхідно атакувати. Необхідно пошкодити одні й ті ж дані у кожному блоці, а так як кожна зміна стає помітною усім учасникам, то така атака потребує значного обсягу комп’ютерних можливостей, що зупиняє практично будь-якого кіберзлочинця.

Завдяки хешуванню, при зміні чи додаванні рядку, новий та попередній запис залишаються видимими, так як жодна інформація не може бути стертою. Виникає ще одна проблема: кожен крок атаки видно всім учасникам

блокчейну. Для кібератаки необхідно непомітно увійти у мережу, зробити свої дії і так само непомітно піти. Враховуючи ще й складність доступу до кожного блоку, дане питання практично нездійснення для будь-якого кіберзлочинця.

Деякі організації почали впроваджувати технологію Blockchain у свою роботу та уже мають значні позитивні результати. Блокчейн стає підмогою для сервісів, користувачі яких переживають про збереження даних: IoT, юриспруденція, медицина, страхування та ін.

Компанія IBM створила хмарний сервіс для тестування додатків в захищеному середовищі. За словами представників IBM, розробники можуть запустити власний блокчейн-sandbox за 12 секунд. Через хвилину після цього він буде готовий до запуску тестових програм.

Ще один проєкт в цій сфері - рішення Enigma від MIT. Enigma дозволяє запускати будь-який код на зашифрованих даних, при цьому роблячи їх «недосяжними» для третьої сторони.

Інші компанії теж шукають застосування блокчейну в сфері безпеки. Наприклад, Humaniq, яка планує використовувати блокчейн спільно з технологіями штучного інтелекту і розпізнавання біометрії для створення сервісу ідентифікації особистості. Рішення виявиться корисним при купівлі товарів в магазинах, оформленні страховок та інших буденних речах. Схоже рішення для авторизації пропонує компанія Remme.

Проєкт Guardtime використовує закриті блокчейни і замінює цифрові підписи RSA підписами KSI. Вони використовують криптографію з хеш-функціями. Компанія сподівається, що це дозволить уникнути проблем в майбутньому, коли квантові комп'ютери отримають поширення. Вона проаналізувала, що її блокчейн KSI ініціює помітні поліпшення в роботі естонського уряду, наприклад, в сфері морських перевезень і страхування, які вимагають серйозної реструктуризації [8].

1.4.1. Практичне застосування технології Blockchain у цілях забезпечення безпеки

На даний момент є ряд існуючих проблем, вирішення яких знайшли за допомогою використання блокчейну. Серед них виділяють [9]:

— Захист від фішингу. Популярність фішингу серед кіберзлочинців викликана його високою ефективністю та відносно малими затратами. Людські слабкості стали основним вектором сучасних кібератак, головне – переконати жертву в легітимності того, що відбувається. Особи починають довіряти листу, що надіслали шахраї чи сайту, тоді проводиться кібератака. Підтвердити дійсність листа можна різними способами. Існують блокчейн-рішення від MetaCert та CloudPhish, що використовують розподілені реєстри для класифікації фішингових та легітимних Uniform Resource Locator (URL). Перевірка URL за допомогою розподіленого реєстру дозволяє швидко розпізнати фішингові ресурси, однак при цьому слід враховувати, що шахрайські сайти і домени живуть від кількох годин до кількох днів, що знижує ефективність такого роду захисту. Основним аргументом за використання блокчейну для захисту від фішингу є його децентралізованість. Завдяки цьому, шахраї не зможуть заблокувати сервер, на якому зберігається база легітимних ресурсів.

— Безпека інтернет речей (Internet of Things, IoT) [10]. Число пристроїв Інтернету речей постійно зростає та їх безпека стає критично важливою. Відсутність криптографічного захисту та вразливість програмного забезпечення роблять IoT ідеальною мішенню для хакерських атак. Використання блокчейну вирішує багато проблем, що пов’язані з інтернетом речей, наприклад, проблему з автентифікацією та підключеннями. Реєстрація

кожного IoT-пристрою у розподіленому реєстрі та видача або ж виключення прав доступу за допомогою блокчейн-транзакцій дають можливість усім учасникам мережі перевірити легітимність підключень та запитів. У результаті, підключення неавторизованих пристроїв і перехоплення або підміна даних стають неможливими. Саме так працює блокчейн-платформа Uniquid на базі Litecoin, що забезпечує захист від несанкціонованих підключень. Інший напрямок використання блокчейна в IoT – захист ланцюжків поставок. Реєстр дає можливість відстежувати всі стадії виробництва і переміщення компонентів готового виробу, ліків чи продуктів харчування, виключаючи можливість крадіжки або підробки.

— Запобігання атак на відмову в обслуговуванні (distributed denial of service (DDoS)). При розподіленій атаці типу «відмова в обслуговуванні», ціль (зазвичай сервер) піддається атаці з боку декількох заражених комп’ютерних систем для відмови в обслуговуванні, що призводить до зниження швидкості, а в кінцевому результаті до перевантаження чи збою у роботі системи. Замість того, щоб діяти з одного центрального сервера, клієнти платформи Blockchain зможуть використовувати колективну силу в об'єднаній мережі проти DDoS атак. Фактично, щоб зламати одну систему, доведеться подолати загальний захист. Щоб проникнути в один з децентралізованих вузлів, DDoS-атаці потрібно відключити всі децентралізовані сервери. Стартап Gladius пропонував використовувати блокчейн для захисту таких атак. Система, яку вони розробляли, повинна була забезпечити децентралізацію смуги пропускання для пом'якшення атак. Передбачалося створення захисту на базі невикористаної смуги

пропускання учасників мережі, за допомогою розподіленого реєстру.

1.4.2. Проблеми впровадження технології Blockchain

Компанії стикаються з багатьма проблемами і питаннями при розгляді можливостей використання технології Blockchain. Однією з найбільших проблем є вартість усього процесу. Більшість існуючих платформ є досить неефективними з точки зору швидкості транзакцій та енергоспоживання, а більшість вимог до програмного забезпечення потребують значних коштів. Крім того, фундаментальні технології Blockchain не готові і неперевірені для великомасштабної комерційної реалізації [11].

До винайдення технології, більшість речей, які блокчейн може змінити, працювали досить добре, не дивлячись на ряд проблем. Зокрема, галузі звикли до усталених структур, а також до стандартів, встановлених відповідно до цих структур. Проблема полягає у тому, що для успішного введення нових технологій необхідно створювати нові структури для управління новим способом, а також необхідно мати достатньо знань, перекваліфіковувати своїх працівників, або ж наймати нових.

Блокчейн організовує проведення операцій таким чином, що користувачам не потрібно участь посередників. Крім того, платформа забезпечує прозорість, а також цілісність даних таким чином, що кожен учасник на платформі може слідкувати за ними. У деяких випадках компаніям необхідно працювати разом, спільно використовуючи бази даних та процеси. Компанії стають вимушеними надавати цінну інформацію, яка за рахунок даної технології може стати доступною для конкурентів та втратити свою конфіденційність. Це стає дуже актуальним, враховуючи, що більшість потенційних прихильників цієї технології знаходяться в галузях, що характеризуються жорсткою конкуренцією та великою секретністю.

Однією з постійних перешкод для реалізації блокчейна є нездатність користувачів на одній платформі досить добре взаємодіяти з користувачами на інших платформах. Ця відсутність взаємодії стримує багатьох потенційних користувачів.

Технологія Blockchain знаходиться на стадії вирішення цих проблем та можливо вже найближчим часом представить свої нові можливості. Проте, також існує ймовірність того, що дані питання так і залишиться не вирішеними та технологія не виправдає своїх очікувань.

1.5. Формалізація постановки задачі дослідження

Задачею першого розділу бакалаврського проєкту є дослідження актуальності проблем, що стосуються забезпечення інформаційної безпеки, ризиків, що можуть спричинити дані проблеми, оцінка існуючих методів для боротьби з загрозами, а також дослідження проблем впровадження нових технологій.

Окрім того, метою даного дослідження є вивчення теоретичного підґрунтя описаних систем, проведення аналізу впливів різноманітних факторів, аналіз будови архітектури, технологій, платформ та існуючих рішень та розробка на цій основі смарт-контракту з використанням технології Blockchain.

2. ТЕХНОЛОГІЯ BLOCKCHAIN ТА СМАРТ-КОНТРАКТИ У МЕРЕЖІ ETHEREUM

2.1. Технологія Blockchain.

Блокчейн – це один з видів розподіленого збереження даних, що використовує уже раніше відомі технології, такі як однорангові мережі (peer-to-peer або P2P), шифрування та бази даних.

Оснований на топології P2P, блокчейн являється технологією розподіленої бухгалтерської книги DLT (distributed ledger technology), яка дозволяє будь-якому користувачу мережі бачити записи інших учасників у реальному часі та створювати власні незмінні записи транзакцій. Це децентралізована мережа, що складається з мільйонів комп'ютерів, які зазвичай називають «вузлами» [1].

Кожна транзакція пов'язана з попередньою та має мітку часу. Щоразу, при додаванні набору транзакцій, дані стають ще одним блоком в ланцюжку, звідки технологія і отримала свою назву. Блоки, що складають ланцюжок між собою – пов'язані (рис. 1).

Рисунок 1 – Ланцюг блоків Blockchain

Блокчейн може бути оновлений тільки на основі консенсусу між учасниками системи, після введення нові дані стерти неможливо [12].

Інформація передається через розподілену мережу комп'ютерів, створюючи так звану розподілену мережу довіри.

Головною особливістю являється відсутність єдиного серверу, усі ланцюжки розподілені між користувачами. Застосування сучасних алгоритмів шифрування дозволяє захищати окремі записи, що належать конкретній особі від копіювання чи редагування іншими користувачами системи. Безпека на рівні бази даних була закладена у блокчейн при створенні. Доступ до інформації має кожен учасник, але жодні особисті дані не зберігаються та важливі лише паролі доступу, систему ніхто не контролює, у всіх рівні права та кожен самостійно відповідає за власні транзакції, які не можна змінити чи скасувати, якщо інформація записана, то вона зберігається та стає частиною ланцюга.

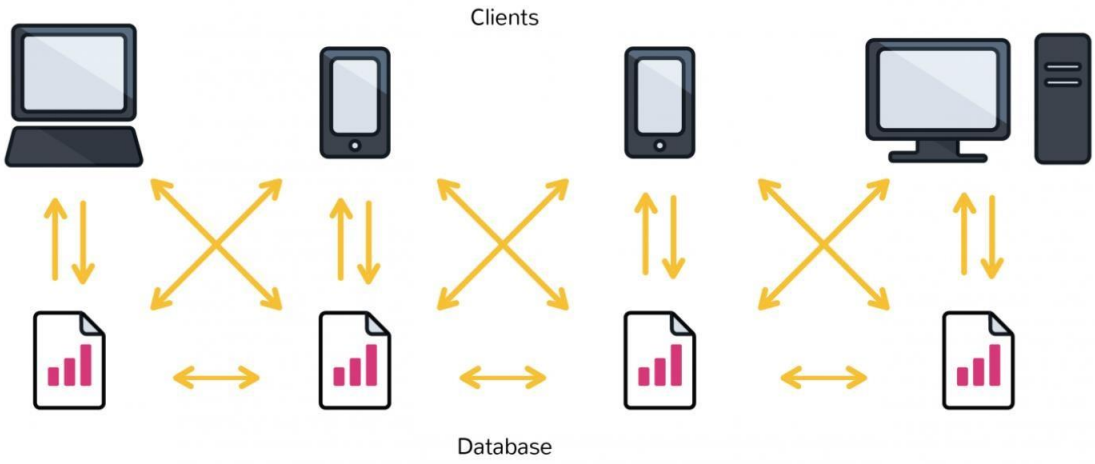

Рисунок 1 – Взаємодія пристроїв у мережі

Ідея створення ланцюжків блоків належить особі, чи групі осіб, що в 2008 році представили себе, як Сатоші Накамото (Satoshi Nakamoto). Вперше це було реалізовано у 2009 році, як компонент цифрової валюти Bitcoin, де блокчейн відіграє роль головного регістру всіх операцій з даною криптовалютою [1].

У потоці блокчейну обмін даних називається транзакцією. Коли відбувається нова транзакція чи модифікація існуючої транзакції, більшість

вузлів у мережі Blockchain здійснюють алгоритми для оцінки та перевірки цього окремого блоку. Якщо транзакція проходить перевірку, вона додається у вигляді нового блоку у базу даних (рис. 2) [12].

Для додавання нового блоку, необхідно отримати відповідь на криптографічну хеш-функцію, яка дійсно являється складною математичною проблемою, яку необхідно вирішити. Для вирішення такої задачі в середньому необхідно близько 10 хвилин, після чого вузол, отримає право додати новий блок у ланцюжок [13].

Як тільки транзакція завершена, вона створює безпечний і унікальний хеш-код з використанням криптографічної техніки хешування, яка зв'язує транзакцію з наступним блоком. Таким чином, блоки розміщені один за одним, щоб організувати ланцюжок пов'язаний з часом.

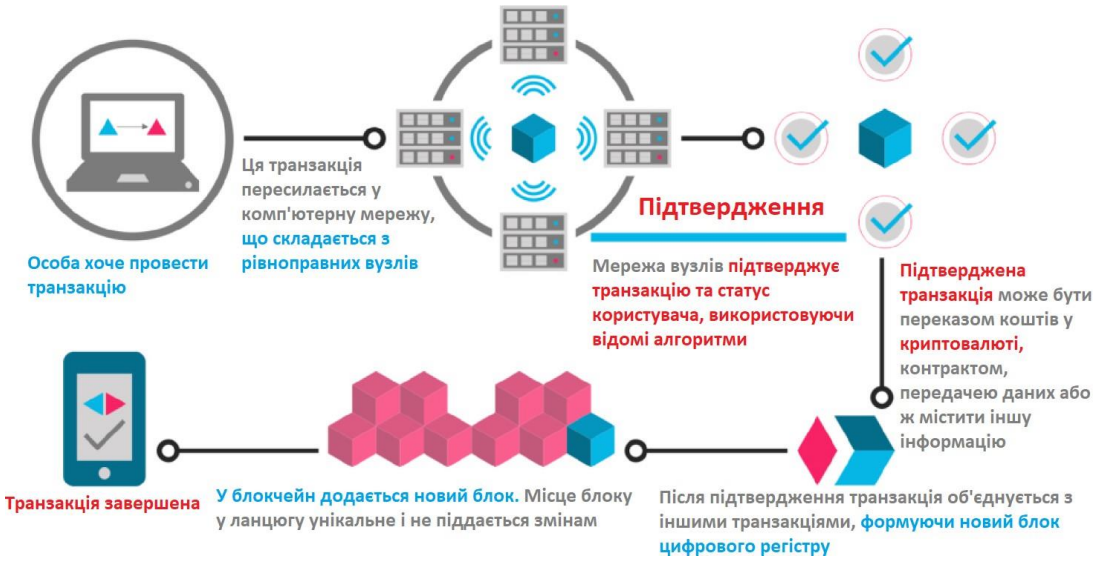

Рисунок 2 – Проведення транзакцій у мережі

Якщо більшість блоків забороняють введення нової чи зміненої транзакції, вона не буде додана у базу даних.

2.1.1. Алгоритм хешування

Функція хешування або ж хеш-функція являється детермінованою функцією, на вхід якої подається рядок бітів довільної довжини, а виходом завжди є бітова послідовність фіксованої довжини n. Хеш це значення хеш-функції H(m) для входу m [13].

Для криптовалют хешування – це перетворення масиву даних будь-якого об’єму у вихідну строку фіксованої довжини. Транзакції, що являються основним масивом даних для блокчейну, проходять через алгоритм хешування (в основу криптовалюти Bitcoin покладено алгоритм SHA-256), який у результаті видає вихідну строку фіксованої довжини (рис. 3).

Input	Output (Hash)
Hi	3639efcd08abb273b1619e82e78c29a7df02c1051b1820e99fc395dcaa3326b8
Nice to meet you	b1e91f811d3e926d1d94b387e41ecdb8d1adfac49c0f9747ee27dfb5cbc331c2

Рисунок 3 – Приклад хешування

Незалежно від довжини і об’єму вхідної послідовності, на виході завжди отримується рядок довжиною 256 біт. Це відіграє значну роль, коли необхідно перетворити в хеш великі і дуже великі масиви інформації, як наприклад, безліч транзакцій в мережі Bitcoin або Ethereum.

Криптографічні хеш-функції мають певні особливості, які дозволяють зробити їхнє використання у криптовалютах безпечним та надійним.

Раніше вже вказувалось, що незалежно від розміру вхідної послідовності, на виході отримується хеш одного і того ж розміру. Дана властивість відіграє важливу роль, оскільки в протилежному випадку неможливо було би відслідковувати вихідні дані.

Хеш-функція швидко повертає вихідні дані, цим самим являється дуже ефективною та актуальною для популярних криптовалют, у блок яких входить велика кількість транзакцій.

Функція стійка до атаки пошуку прообразу. Суть цієї властивості полягає у тому, що якщо значення H(A) відоме, де A – вхідна послідовність, а H(A) – вихідна послідовність, тобто хеш, то знаходження значення A являється практично нездійсненною задачею [13].

Ще однією властивістю є стійкість до колізії. Ймовірність знаходження двох вхідних послідовностей, які б мали одну і ту ж вихідну послідовність після проходження процесу хешування, повинна бути максимально наближена до 0. Теоретично, це може бути можливим, але час, що витрачається на знаходження двох однакових прообразів виміряється десятками років. Дана властивість відіграє важливу роль у питаннях безпеки криптовалют.

Одну з найважливіших ролей у забезпечення безпеки та надійності блокчейну відіграє властивість лавинного ефекту. Внесення навіть незначних змін, наприклад – зміна регістру, у вхідну послідовність призводить до кардинальної зміни хешу (рис. 4).

Input	**Output (Hash)**
blockchain and cryptocurrencies	dd5f90aea0aa651c40a8f1d2778898ea872666360599d7375f4b60a4f1f2317d
Blockchain and Cryptocurrencies	697ab64a181fd8b5aec9dc40a125aac2e59e04bb586170a31128340de1683b86

Рисунок 4 – Приклад хешування

Хеш-функція має високий показник ентропії, за рахунок чого функція забезпечує складність підбору вхідних даних для зловмисників, шанс знайти правильний Input дуже низький.

2.2. Блоки у Blockchain

Блок це базова складова блокчейну, це файли, що записуються без майбутніх змін у мережі. Вони містять інформацію про проведені транзакції до моменту створення даного файлу. Окрім того, у блоці записуються усі нещодавні транзакції (повністю або частково), що не входили у попередні блоки. Кожен новий блок додається в кінець ланцюга блокчейну. Усі блоки

пов’язані між собою та кожен наступний блок містить інформацію про попередні блоки, а також формується ланцюжок блоків.

Формуються нові блоки за допомогою майнінгу. Майнінг – це спосіб підтримки розподіленої платформи і створення нових блоків, шляхом пошуку певного числа. Транзакції, що проводяться всередині мережі, записуються у блоки, а потім більшість майнерів перевіряють їх дійсність і проводять обчислення над блоком. Якщо всі дані вірні, то майнер, що швидше всіх закінчить обчислення закриває блок та отримує винагороду. Для вирішення даної задачі необхідно задіяти потужні ресурси обчислювальної системи. Кожна наступна задача стає все важчою, у міру зростання кількості блоків у мережі. Єдиного правильного рішення не існує, кожен блок можна обчислити декількома способами [14].

Перший блок називається генезис-блоком, блоком #0, первинним блоком або ж блоком буття, з нього починається будь-яка криптовалюта.

У блокчейні немає максимальної кількості блоків, а загальна кількість блоків, що з'єднані з генезисним блоком, називається висотою блоку. Блоки підраховуються послідовно, висота вказує на розташування блоку в блокчейні, а найвищий блок являється останнім блоком та вказує на загальну висоту блокчейну.

Висота блоку є хорошим показником часу та використовується для вимірювання зрілості та відстані мережі Blockchain, але висота блоку (або ж його номер) не записується у інформацію, що містить блок. Лише вузли мережі відслідковують цей номер, однак, кожен блок містить часову мітку для забезпечення цілісності блокчейну (рис. 5).

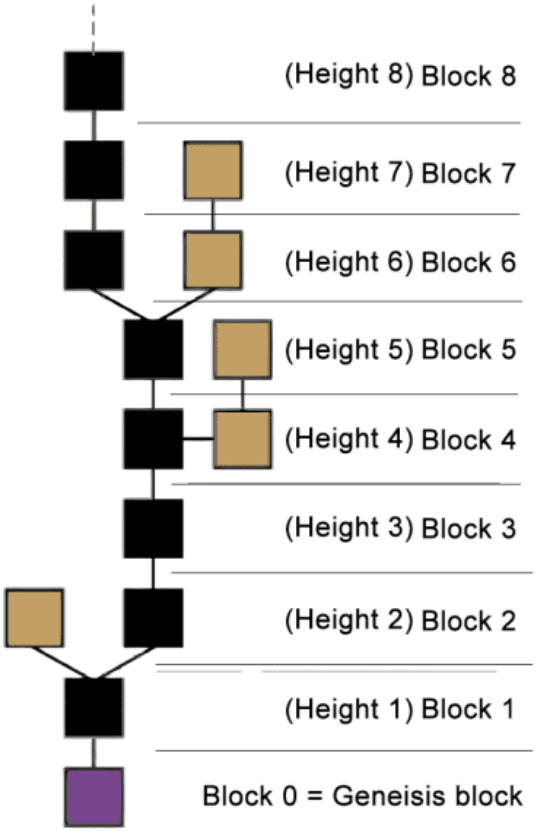

Рисунок 5 – Висота блоку

Блоки містять величезну кількість даних та множину складових. Кожен блок починається з «заголовку», у якому присутні:

— власний хеш;

— хеш попереднього «батьківського» блоку;

— хеш проведених транзакцій;

— службова інформація з додатковими даними.

Окрім того, у блок вноситься інформація про кожну записану транзакцію, таким чином [15]:

1. Прораховуються загальні хеші всіх транзакцій у даному блоці.
2. Розраховуються хеш-функції від суми хешів цих транзакцій.
3. По такій ж схемі розраховуються хеші від отриманих хеш-функцій.
4. Розрахунки проводяться до отримання одного загального хешу.

Таким чином формується хеш-дерево або дерево Меркла. Хеш-дерево – це двійкове дерево, кінцеві вузли якого – це хеші транзакцій, а внутрішні вершини – результати складання значень пов'язаних вершин (рис. 6). За допомогою хеш-дерева можна перетворити величезний обсяг даних у один єдиний хеш [15].

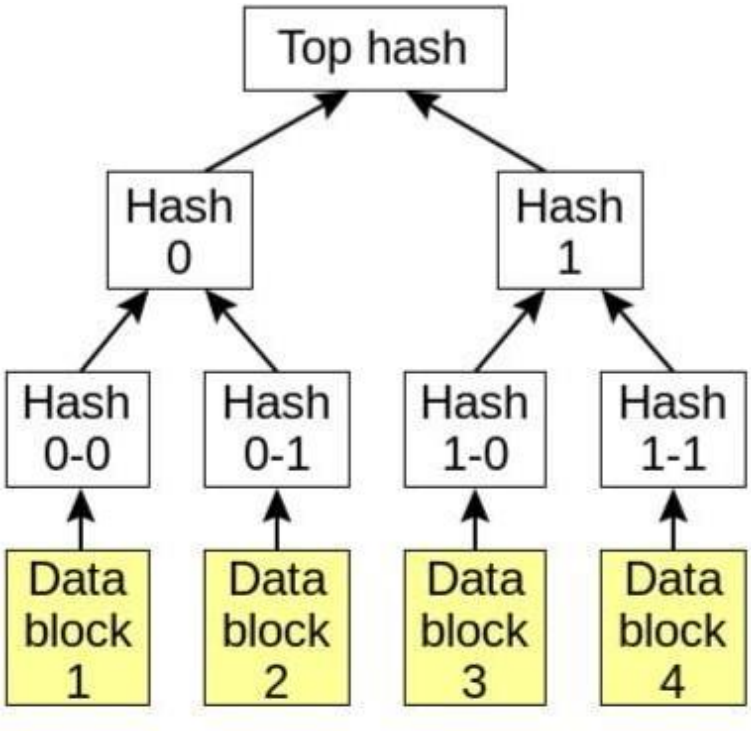

Рисунок 6 – Хеш-дерево

2.3. Типи блокчейну

Технологія постійно розвивається, а разом з нею з'являються і нові види блокчейну, їх класифікують у залежності доступу до реєстру даних. Сам по собі розподіл на види є лише умовністю, оскільки, технологія залишається тією ж [1].

Найбільш очевидний розподіл блокчейну – це за ступенем його відкритості.

Ступінь відкритості блокчейну може залежати від декількох факторів. Перший – доступність вихідного коду блокчейну (зазвичай під цим мається на увазі найбільш поширений клієнт даного блокчейну). Для Bitcoin таким клієнтом є Bitcoin Core. А, наприклад, в системі Ethereum немає одного загальноприйнятого клієнта, і користувачі можуть вибирати з декількох варіантів: Geth, Parity, MyEtherWallet і т.д., і всі вони розробляються з відкритим вихідним кодом.

Блокчейн та платформи для корпоративного застосування можуть бути як з відкритим, так і з частково закритим вихідним кодом. Щодо блокчейн-розробок для державних установ, то їх кількість на даний момент досить мала, але слід очікувати, що більшість з них також будуть закритими.

Другий та найбільш важливий показник – це можливість для будь-якого користувача вільного підключення до мережі без будь-яких дозволів. Переважна більшість відомих на сьогодні блокчейнів публічні – для підключення до них достатньо завантажити лише сумісну з поточною версією протоколу програму (клієнт) та встановити зв’язок з іншими рівноправними вузлами мережі. Окрім того, ніхто не може відключити користувача від розподіленої мережі, оскільки, всі учасники публічного блокчейну рівноправні [16].

Спроби знайти баланс між децентралізованістю, безпекою, конфіденційністю та розмежуванням доступу привели до численних експериментів з механізмами управління блокчейном. Існуючі на даний момент блокчейн-системи за рівнем управління можна розбити на групи [16].

Відкритий блокчейн або ж публічні децентралізовані системи (Permissionless Blockchain), які зазвичай є дуже великими. Будь-хто може прийняти участь у їх роботі та на будь-якому рівні підтримувати роботу повного вузла, майнити криптовалюту, торгувати токенами чи записувати нові дані. Ці мережі зазвичай краще забезпечують захист та незмінність, аніж інші мережі, але працюють не так швидко та їх експлуатація в рази дорожча. Вони

маніпулюють захищеною інформацією та мають строгі обмеження на розмір збережених у записі даних. Іншими словами, це повністю децентралізована мережа, у якій не може бути учасників, що мають вищий рівень. Можливості учасників визнаються лише долею наявних у них ресурсів. До даного виду блокчейну відносяться такі криптовалюти, як Bitcoin, Ethereum, Bitcoin Cash чи Litecoin.

У приватних блокчейнах за підключення до мережі нових користувачів (а також за можливість їх відключення) можуть відповідати виділені довірені вузли чи групи вузлів, що мають вищий рівень повноважень у порівнянні з іншими користувачами. Приватні блокчейни представляють собою ієрархічні структури, що складаються з двох або більше рівнів. Проте, вони не реалізують основні принципи технології – децентралізацію та рівноправність учасників, так як для корпоративних систем їх наявність несе за собою істотні ризики.

У 2015 році виникли перші публічні блокчейни з дворівневою структурою, де основну роль відігравали так звані суперноди, або вузли з розширеними повноваженнями. Це відкритий блокчейн з різним рівнем дозволів (Public Permissioned Blockchain). У цьому виді транзакції підтверджують визначені люди. Це може бути керівний орган, старший співробітник, уряд, установа чи хтось ще, а користувачі можуть переглядати дані (особливо важлива інформація може бути прихована). Такі ланцюжки блоків мають вже централізовану структуру.

З одного боку, багато користувачів даної структури обмежені у використанні, а прозорість угод не забезпечується. Проте, з іншого боку, для деяких завдань у такій організації мережі є переваги перед повністю відкритим блокчейном, наприклад, для корпоративних мереж, що захищають приватну інформацію клієнтів.

Приватний блокчейн (Private Permissioned Blockchain) схожий на публічний блокчейн, але є одне виключення. Доступ до записів реєстру має

тільки обмежене коло осіб. Він може бути як децентралізованим, так і централізованим, де в якості основного валідатора виступає конкретний суб'єкт. Головною перевагою є те, що можна уникнути зайвого навантаження на мережу, за рахунок чого транзакції будуть проходити швидко, а їх деталі будуть приховані від третіх осіб. Сам реєстр буде мати невеликий розмір за рахунок того, що записів про транзакції буде небагато, та обробка проходитиме швидше, а конкретні відомості можна знайти простіше.

Класичним прикладом є платформа Mijin, що була створена на основі криптовалюти NEM. Приватні блокчейни найкраще підходять для корпоративних мереж та можуть полегшити перехід фінансових операцій у криптоіндустрію.

2.4. Переваги та недоліки технології

Створення технології принесло множину переваг для різних сфер, забезпечуючи безпеку в надійних умовах.

До переваг відносять [17]:

— Розподілені дані. Дані знаходяться на багатьох пристроях, у розподіленій мережі вузлів, система та дані мають високу стійкість до технічних збоїв та зловмисних атак (рис.7). Кожен мережевий вузол може копіювати та зберігати копію бази даних. Не існує єдиної точки відмови, один відключений вузол не впливає на доступність чи безпеку мережі. На відміну від блокчейну, багато баз даних використовують один або декілька серверів, і тим самим є більш вразливими до кібератак та технічних збоїв.

— Стабільність. Після реєстрації даних у блокчейні блоки неможливо змінити чи видалити. Дана перевага робить блокчейн відмінною технологією для збереження записів, котрі потребують журнал звітності, оскільки, кожна зміна

відслідковується та постійно записується у загальнодоступний регістр. Наприклад, для бізнесу, технологія блокчейн може використовуватись, щоб запобігти шахрайству зі сторони співробітників. Це може забезпечити безпечний та стабільний запис усіх транзакцій.

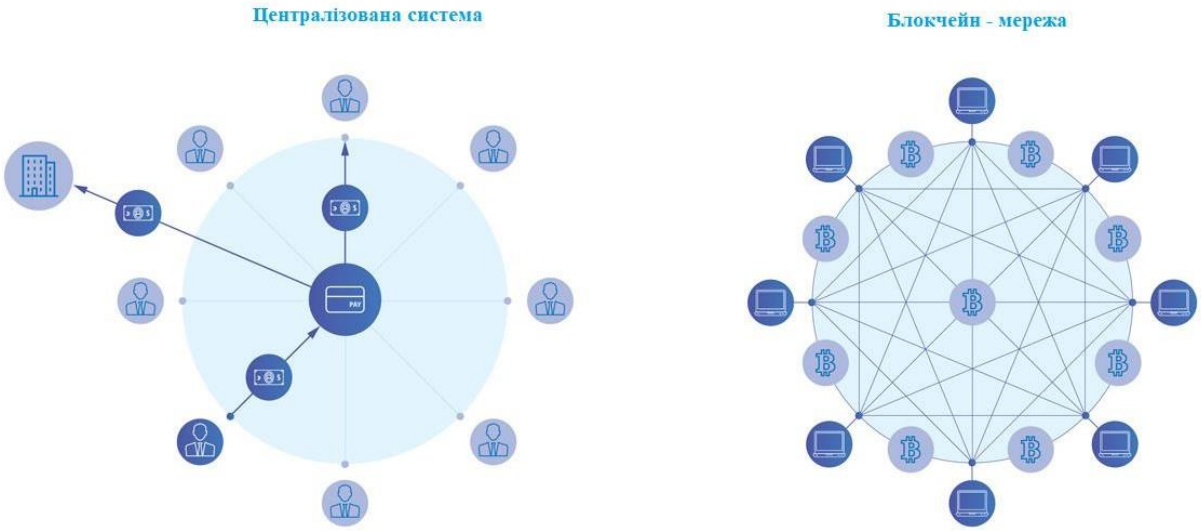

Рисунок 7 – Розподілена блокчейн мережа

— Система довіри. У більшості традиційних систем, транзакції залежать не лише від двох зацікавлених сторін, але і від посередника, компанії-емітента платіжних карт чи поставника платежів, банку. При використанні технології блокчейну це більше не потрібно, оскільки розподілена мережа вузлів перевіряє транзакції за допомогою процесу, відомого як майнинг. Таким чином, блокчейн система зводить практично до нуля ризик довіри до однієї організації, а також зменшує загальні комісійні витрати на транзакції, виключаючи посередників та третіх осіб.

— Універсальність. Блокчейн дозволяє створювати загальнодоступні бази даних: земельні реєстри, відкриті

ресурси для реєстрації прав власності, у тому ж числі інтелектуальної, управління енергетичними потоками, голосування через інтернет та багато іншого.

— Смарт-контракти. Це невеликий програмний файл, що описує взаємодію користувачів, які діють без посередників. Все контролюється з використанням попередньо написаної програми. Людський фактор зникає, що надає важливі переваги.

Проте, не дивлячись на усі переваги, децентралізований характер блокчейну також має деякі недоліки. До них відносять [17]:

— Атака 51%. Існує декілька потенційних атак, які можуть бути здійснені на блокчейн мережу, атака 51% являється однією з найбільш обговорюваних. Така атака може відбутись, якщо одному об'єкту вдасться контролювати більше 50% потужності хешування мережі, що в підсумку дозволить порушити її роботу, шляхом навмисного виключення або зміни порядку транзакцій. Не дивлячись на те, що це теоретично це можливо, атака 51% ніколи не була успішною. У міру зростання мережі збільшується її безпека та малоймовірним стає те, що майнери будуть вкладати величезні суми грошей і ресурсів на атаку мережі.

— Модифікація даних. Дуже важко модифікувати дані, після додавання у блокчейн ланцюжок. Хоча і стабільність вважається перевагою технології, проте, це не завжди добре. Зміна даних чи коду у блокчейні, як правило, потребує значних зусиль та найчастіше для цього необхідний хардфорк (зміна криптовалютного протоколу), коли один ланцюг залишається та починає використовуватись новий.

— Приватні ключі. Блокчейн використовує криптографію з публічними ключами (асиметричне шифрування), щоб

представити користувачам право власності на свою частину криптовалюти (чи будь-які інші дані в блокчейні). Кожен акаунт у блокчейні (чи адреса) мають два відповідних ключа: відкритий ключ (яким можна поділитися) і приватний ключ (який повинен зберігатись у секреті). Користувачам потрібен приватний ключ для доступу до своїх даних. У разі його втрати, доступ втрачається та не підлягає відновленню.

— Неефективність. Блокчейни, особливо ті, що використовують алгоритм Proof of Work, вкрай не ефективні, за рахунок висококонкурентної системи майнінгу. Кожні десять хвилин виграє лише один майнер, а робота кожного другого втрачається. Майнери постійно намагаються збільшити свою обчислювальну потужність, у них з'являється більше шансів знайти дійсний блок, хеш чи ресурси, що використовує мережа.

— Сховище. Блокчейн регістри з часом можуть стати дуже об'ємними. Bitcoin блокчейн в даний час вимагає близько 200 ГБ пам'яті. Поточне зростання розміру блокчейна, випереджає зростання розмірів жорстких дисків, і тому мережа ризикує втратити вузли, якщо регістр стане занадто великим для завантаження і зберігання користувачами.

2.4.1. Переваги Blockchain для кібербезпеки

Однорангова мережа, розподілений консенсус та шифрування являються основними принципами для забезпечення безпеки на основі Blockchain. Нова технологія може протистояти ряду сучасних кіберпроблем. Основними перевагами для забезпечення кібербезпеки являються [17]:

— Децентралізація. Завдяки одноранговій мережі немає необхідності у сторонній перевірці, так як будь-який користувач може бачити мережеві транзакції.

— Відстеження та контроль. Усі транзакції у ланцюжках блоків мають цифровий підпис, а також позначку часу. За рахунок цього, учасники можуть легко відстежувати історію проведених операцій та облікові записи у будь-який момент часу.

— Конфіденційність. Конфіденційність учасників здійснюється завдяки криптографії з відкритим ключем для перевірки учасників і шифрування їх операцій.

— Право на забуття. Після реєстрації у блокчейні інформація стає унікальною. Це гарантує безпеку даних, дозволяє системі захищати себе від будь-яких незаконних або дублюючих транзакцій, запобігає можливість видалення інформації.

— Стійкість. Технологія Blockchain не має єдиної точки збою. Це означає, що навіть у разі DDoS-атак система буде працювати в звичайному режимі завдяки безлічі копій цифрових даних.

— Цілісність. Розподілений регістр забезпечує захист даних від несанкціонованого змінення або знищення. Крім того, технологія забезпечує достовірність і незворотність завершених угод.

— Гнучкість. Використання мережі гарантує, що Blockchain буде працювати цілодобово, навіть якщо деякі вузли знаходяться в автономному режимі або під впливом атаки.

2.5. Алгоритми консенсусу у Blockchain

Консенсус у Blockchain – це процес досягнення угоди з питання значення розподілених даних - необхідно, щоб прийняте рішення було одним.

Алгоритми консенсусу складають основу технології блокчейн. Їх роль полягає у досягненні рівня надійності мережі, побудованої на серії вузлів (пристроїв, з'єднаних з іншими пристроями як частина комп'ютерної мережі). Це означає, що, якщо здійснена транзакція, то алгоритм почне працювати – обмінюватись інформацією в мережі, щоб перевірити, чи може дана дія мати місце. Такий алгоритм також застосовується для створення нових вузлів даних у блокчейні або при синхронізації мережевого обладнання, щоб забезпечити узгодженість всього консенсусу.

Не всі блокчейни створюються однаковими, і розрізняються залежно від типу алгоритму консенсусу, який вони використовують. Найбільш поширеними алгоритмами є Proof-of-Work (PoW) та Proof-of-Stake (PoS). У кожного з них є свої переваги і недоліки, при спробі збалансування безпеки, з функціональністю і масштабованістю [18].

Proof-of-Work є першим створеним алгоритмом консенсусу. Він використовується для підтвердження транзакцій і створення нових блоків. У мережі користувачі надсилають один одному цифрові токени. Децентралізований регістр об'єднує всі транзакції в блоки, однак, слід подбати про те, щоб підтвердити транзакції і впорядкувати блоки. Ця відповідальність лежить на спеціальних вузлах, названих майнерами, а сам процес називається майнінгом.

Ця проблема потребує значних обчислювальних ресурсів. Робота мережі заснована на вирішенні складних математичних задач і можливості легко довести, що рішення отримано. Таких математичних задач є декілька, це є вирішення хеш-функцій, а саме спроби знайти вхідні дані, знаючи вихідні, розкладання цілого числа на множники, та інші.

У випадку з PoW використовується хешування. У міру зростання мережі проблеми стають все серйозніше, і алгоритми хешування вимагають все більшої обчислювальної потужності, так що складність завдання - актуальна проблема.

Від цього механізму залежить точність і швидкість блокчейна. При цьому проблема не повинна бути занадто складною - у цьому випадку генерація блоку займе багато часу, а значить, в мережі «зависне» багато незавершених транзакцій. Якщо проблема не може бути вирішена за передбачуваний час, створення блоків стане лише випадковістю. Якщо ж проблема вирішується дуже просто, це робить систему вразливою для зловживань, спаму і DDoS-атак. Рішення повинно легко перевірятись, в іншому випадку не всі вузли зможуть зрозуміти, чи правильно був проведений розрахунок, а значить, їм доведеться довіряти іншим вузлам, що не узгоджується з одним із найважливіших принципів блокчейна - повною прозорістю.

Майнери вирішують задачу, формують новий блок і підтверджують транзакції. Складність завдання залежить від кількості користувачів, поточної потужності і навантаження на мережу. Крім того, хеш кожного блоку містить в тому числі хеш попереднього блоку, що підвищує безпеку і унеможливлює порушення порядку створених блоків. Якщо майнер зумів вирішити завдання, формується новий блок - в ньому розміщується черговий комплект транзакцій, і вони вважаються підтвердженими.

Алгоритм PoW використовується у більшості криптовалют, у тому числі і у валюті Bitcoin, і саме ця валюта почала першою використовувати даний алгоритм. Тут використовується алгоритм Hashcash, що дозволяє змінювати складність завдання у залежності від загальної обчислювальної потужності мережі. Подібна ж система реалізована в схожих криптовалютах, наприклад, у валюті Litecoin.

Основними перевагами алгоритму є захист від DDoS-атак і низький вплив частки криптовалюти у власності у майнера на можливості видобутку. PoW накладає певні обмеження на дії учасників, оскільки для вирішення завдання потрібні значні зусилля. Ефективна атака також вимагає великих і тривалих обчислень, тому вона можлива, але невигідна на тлі високих витрат.

Неважливо, скільки грошей у вас в гаманці - важливо мати великі обчислювальні можливості для вирішення завдань і формування нових блоків, а значить, власники великих капіталів не можуть приймати рішення за всю мережу.

Недоліками є величезні витрати, деякі марні обчислення та «атака 51%». Для складних розрахунків потрібне спеціалізоване і дороге комп'ютерне обладнання. Витрати ростуть з некерованими, і майнінг стає можливий тільки для великих груп майнерів. Крім того, спеціалізовані комп'ютери споживають масу енергії, що збільшує витрати. Наслідком цього стає поступове підвищення централізації системи, оскільки, це вигідно. І саме це відбувається у випадку з біткоінами. Майнери виконують роботу зі створення блоків, попутно споживаючи величезну кількість енергії, але обчислення, які вони роблять, абсолютно марні самі по собі. Так, вони гарантують безпеку в мережі, але їх результати не можна використовувати деінде. «Атака 51%» або атака більшості можлива в ситуації, коли користувач або група користувачів контролюють більшу частину мережі - це дає їм можливість контролювати події, що відбуваються у ній. Саме так, вони можуть монополізувати створення нових блоків і отримувати всі винагороди, оскільки вони перешкоджають іншим майнерам завершувати блоки.

Алгоритм Proof-of-Stake працює по-іншому. PoS замінює PoW з метою уникнення марної трати ресурсів на майнінг. Замість цього всі вхідні параметри строго задані з константною характеристикою на основі існуючих заощаджень власників монет [18].

Ця модель заснована на ідеї, що чим більше хтось вкладає в систему, тим менше він хоче її обдурити. Ставка - це кількість криптовалюти, яку користувач мережі блокчейн вклав в систему, тому одного разу поставлена криптовалюта більше не може бути витрачена.

Учасники, які мають значну частку в системі, вибираються псевдовипадковим чином для створення і подальшого додавання блоків в

ланцюжок блоків. Цей псевдовипадковий вибір відбувається після аналізу декількох різних факторів, щоб гарантувати, що вибираються не тільки люди з величезною часткою, а й інші. Деякі з цих факторів вибору - рандомізований вибір блоків, вибір монет за віком, мастерноди і т.д. POS зазвичай застосовується до тих криптовалютат, які попередньо видобуваються, щоб користувачі мали доступ до монет для розміщення. Це означає, що поставка POS-криптосервісів фіксована з самого початку, і немає ніякого блочного майнінгу або винагороди за створення, як у алгоритмі PoW. Таким чином, єдиним стимулом, який отримують PoS-фальсифікатори, є плата за транзакцію, прикріплена до блоку.

Очевидна перевага Proof-of-Stake перед Proof-of-Work полягає в тому, що його виконання не вимагає від майнера величезних витрат електроенергії, що робить його більш ефективним. Але це не єдина його перевага. Він також дозволяє реалізувати системи управління нового покоління. При використанні механізму PoS немає сенсу проводити «атаку 51%», оскільки, для її здійснення необхідно мати величезну кількість монет, а у разі атаки, шахрай першим ж постраждає від неї.

Недоліками є те, що володіння монетами дає додаткову мотивацію до накопичення коштів в одних руках, що може призводити до централізації мережі. Якщо утворюється невелика група, яка сконцентрує у себе чималі кошти, вона зможе нав'язувати свої умови функціонування криптовалюти, з якими будуть незгодні інші учасники мережі.

Proof-of-Work і Proof-of-Stake можна вважати двома найбільш популярними алгоритмами консенсусу в світі криптовалюти, однак крім них, існує ще цілий ряд механізмів, що мають власні особливості.

2.6. Смарт-контракти

Смарт-контракти або ж криптоконтракти призначені для укладення та виконання договорів. Це програми, що написані спеціально для автоматизації

управлінні передачі активів між двома чи більше сторонами, у разі виконання певних умов. На базовому рівні ці програми працюють так, як їх створили розробники. Смарт-контракти активуються автоматично, коли будуть виконані певні умови. Ця ідея усуває потребу в сторонніх компаніях. При здійсненні транзакцій більше не потрібна довірена третя сторона. Натомість договори здійснюються самостійно у надійній мережі, що повністю контролюється комп'ютерами.

Смарт-контракти є досить складними, їх функціонал не обмежений простою передачею активів. Вони можуть забезпечувати операції у широкому спектрі послуг та перешкоджати юридичній та фінансовій сфері, шляхом спрощення та автоматизації рутинних та повторюваних процесів, за які наразі люди сплачують кошти юридичним та фізичним особам.

Поняття смарт-контрактів з'явилось задовго до появи технології Blockchain, у 1990-х роках його ввів Нік Сабо. Поява поняття була викликана широким поширенням та популярністю інтернету. Автор визначив три основних властивості, що повинен містити смарт-контракт, а саме: спостереження, перевірка та конфіденційність. Всі учасники договору можуть спостерігати кожен етап виконання контракту та перевіряти факт його виконання, а треті особи не повинні впливати на виконання.

У 2009 році ідея була застосована на практиці, завдяки появі технології Blockchain у криптовалюті Bitcoin.

Справа в тому, що концепція блокчейну, використовувана у криптовалютах, передбачає зберігання в кожній операції інформації про всі проведені раніше операції, при цьому такий реєстр зберігається не на одному центральному сервері, а у кожного активного учасника мережі. Все це робить смарт-контракти максимально інформативними (наприклад, можна подивитися інформацію про всіх власників нерухомості з моменту її будівництва) і захищеними від шахрайства (не можна зламати або підробити

сервер з даними, так як інформація про угоди і їх умовах є у всіх активних користувачів системи).

Оскільки, творці криптовалюти Bitcoin обмежили можливість програмування смарт-контрактів, на той час повністю реалізувати концепцію не вдалось.

Активний розвиток смарт-контрактів розпочавсь у 2013 році разом з появою цифрової валюти Ethereum. Її засновник Віталій Бутерін розширив можливості створення смарт-контрактів в середовищі ефіріума, створивши універсальну децентралізовану блокчейн-платформу з можливістю програмування різних систем зберігання і обробки даних. Головна вимога – умови контракту повинні бути описані як математичні правила. Сьогодні до платформ, у яких можуть бути реалізовані смарт-контракти, додалися системи Side Chains і NXT [2].

2.6.1. Принцип роботи смарт-контрактів

Блокчейн Ефіріума, по суті, є машиною станів, що функціонує за допомогою транзакцій. У комп'ютерних науках визначення машини станів означає те, що цей механізм зчитує серію вхідних даних і, ґрунтуючись на них, переходить в новий стан. Саме на цьому блокчейні були побудовані смарт-контракти.

Існує середовище виконання смарт-контрактів – це віртуальна машина Ethereum або ж EVM. Смарт-контракти у даному середовищі вважаються повністю ізольованими, тобто працюючий код не має доступу до мережі, файлової системи та інших процесів, а також мають обмежений доступ до інших смарт-контрактів [19].

Ethereum має два види облікових записів, які використовують один і той же адресний простір: зовнішні облікові записи, які контролюються парами відкритого і закритого ключів, а також контрактні облікові записи, які контролюються кодом та зберігаються разом з обліковим записом (рис. 8).

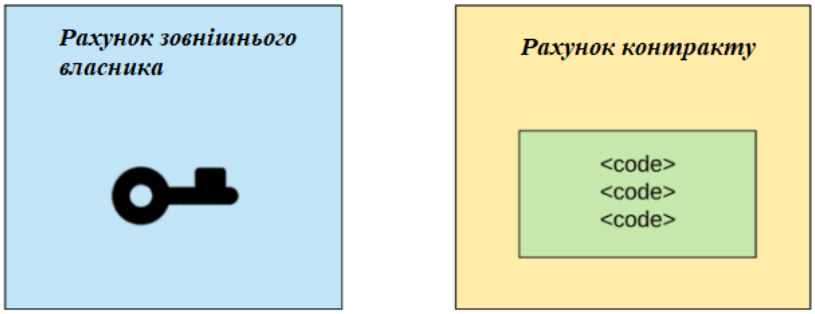

Рисунок 9 – Облікові записи Ethereum

Рахунки зовнішніх власників можуть відправляти повідомлення на адреси інших рахунків зовнішніх власників або рахунків контрактів, створюючи і підписуючи транзакції своїм секретним ключем. Повідомлення, передане від одного рахунку зовнішнього власника іншому - це просто грошовий переказ. Повідомлення ж, відправлене від рахунку зовнішнього власника на адресу рахунку контракту, активує виконання програмного коду контракту, що дозволяє йому виконувати різні дії.

На відміну від рахунків зовнішніх власників, рахунки контрактів не можуть самостійно ініціювати нові транзакції. Вони здійснюють їх тільки у відповідь на отримані транзакції (від рахунку зовнішнього власника або від іншого рахунку контракту).

Таким чином, всі дії в блокчейні Ефіріума ініційовані транзакціями, відправленими з рахунків зовнішніх власників.

Незалежно від того, чи зберігає код обліковий запис, EVM однаково відноситься до двох типів. У кожного облікового запису є постійне сховище значень ключів, що відображає 256-бітові слова в 256-бітові слова, названі сховищем. Крім того, кожен обліковий запис має баланс в Ether (точніше, в «Wei»), який можна змінити, відправляючи транзакції, які включають Ether.

Повідомлення, тобто транзакції, передаються від одного облікового запису до іншого. Вони містять параметри, які використовує смарт-контракт

для коректної роботи. До цих параметрів відносять: адресу отримувача повідомлення, цифровий підпис відправника, для ідентифікації його особистості, суму переказу, поле даних, для відправки повідомлення, а також кількість кроків, дозволених для проведення транзакції та плату відправника за газ.

При створенні кожна транзакція заправляється певною кількістю газу, метою якого є обмеження обсягу роботи, необхідної для виконання транзакції і оплати за це виконання. Gas Limit і Gas Price відповідають за комісію. Ціна кожної операції фіксована в цих одиницях газу і однакова на всіх машинах. Поки EVM виконує транзакцію, газ поступово виснажується відповідно до певних правил. Ціна на газ - це значення, встановлене власником транзакції, який повинен заплатити авансом з відправного рахунку. Якщо після транзакції залишилося трохи газу, він повертається таким же чином. Якщо газ використовується, тобто приймає від'ємне значення, спрацьовує виключення та скасовуються всі зміни, внесені у поточний момент [20].

Творці Ethereum модифікували класичний блокчейн, додавши в нього одну важливу особливість – сховище. Кожна транзакція записує зміни у це сховище. Усі транзакції виконуються у області пам'яті, що має назву стек. Він має максимальний розмір 1024 елемента та містить слова 256 біт. Доступ до нього обмежений, можна переміщати елементи стеку в сховищі, але неможливо отримати доступ до елементів глибше, не видаливши спочатку верхню частину стеку.

EVM підтримує мінімальний набір команд, щоб уникнути неправильних реалізацій, які можуть викликати проблеми консенсусу. Всі інструкції працюють з базовим типом даних, 256-бітними словами. Присутні звичайні арифметичні, бітові, логічні і порівняльні операції, умовні та безумовні стрибки. Крім того, контракти можуть отримувати доступ до відповідних властивостей поточного блоку, таких як його номер і часова мітка.

Контракти можуть навіть створювати інші контракти, використовуючи спеціальний код операції.

2.6.2. Переваги та недоліки смарт-контрактів

Не дивлячись на те, що смарт-контракти мають перспективний потенціал та уже найближчим часом зможуть вирішити та полегшити купу питань, вони мають як і переваги, так і недоліки [21].

До переваг відносять:

— Чіткість умов. Умови контракту прописані у коді та доступні двом сторонам через узгоджене джерело даних.

— Незалежність. Смарт-контракти виключають втручання посередників. Гарантія на транзакцію - сама програма, яка, на відміну від посередників, не дасть підстави сумніватися в її цілісності.

— Швидкість. Обробка документів вручну займає багато часу та затримує виконання задач, смарт-контракти автоматизують процес та економлять дорогоцінний час.

— Надійність. Дані, не можуть бути змінені чи видалені, сторони угоди захищені умовами інтелектуального договору.

— Відсутність помилок. Автоматична система для виконання транзакцій та відсутність людського фактору забезпечують відсутність помилок.

— Економія. Смарт-контракти можуть забезпечити значну економію за рахунок відсутності витрат на посередників.

Основними недоліками є:

— Відсутність правового регулювання. У міжнародно-правовій області відсутні концепції блокчейну, смарт-контрактів та криптовалют.

— Складність реалізації. Хоча, смарт-контракти економлять витрати на посередників, проте, їх інтеграція з елементами реального життя часто займає достатньо багато часу, витрат та зусиль.

— Неможливість зміни. Одну з основних переваг, також можна розглядати як і недолік. Іноді сторони знаходять більш вигідні умови, або ж виникають нові фактори, змінити контракт неможливо.

— Помилки. Сам по собі код може містити помилки, та нести за собою проблеми виконання контракту. На це впливає людський фактор при написанні смарт-контракту.

— Відкритий реєстр. Виникає проблема збереження банківської конфіденційності.

— Оракули [21]. Вважаються основною проблемою смарт-контрактів. Ці програми слугують зв’язком смарт-контрактів з зовнішніми джерелами інформації, тобто виступають як міст між блокчейном та зовнішнім світом. Основна проблема з якою стикаються при розробці оракулів, полягає в тому, що ризик можливих негативних впливів на оракул безпосередньо позначається на роботі смарт-контракту, який на ньому ґрунтується. На жаль, оракули не є частиною механізмів безпеки публічних блокчейнів, так як вони жодним чином не пов'язані з консенсусом блокчейн-мережі. Конфлікт довіри між сторонніми оракулами і ненадійним виконанням смарт-контрактів все ще залишається невирішеною проблемою.

2.6.3. Застосування смарт-контрактів у реальному житті

Потенційні можливості та сфери використання смарт-контрактів величезні – від простого мультипідпису до операцій з похідними фінансовими інструментами.

Мультипідпис (multisig, escrow) – найпростіший, класичний приклад смарт-контракту. З його допомогою сторони, що не довіряють один одному можуть заморозити деяку суму монет у блокчейні таким чином, що у випадку необхідності витратити цю суму будуть необхідні підписи більше ніж половини учасників [22].

Смарт-контракти широко використовуються у сфері первинних розподілів монет (Initial Coin Offering, ICO). Можна запрограмувати смарт-контракт таким чином, що відправляючи криптовалюту на гаманець проекту, учасники будуть впевнені, що в разі провалу кампанії їхні кошти будуть автоматично повернені. Якщо ж фінансова ціль буде досягнута, їхні кошти буде переказано розробникам.

Існує також багато застосувань смарт-контрактів, що можуть знайти місце у щоденному житті. Вони можуть автоматизувати комунальні платежі. Є постачальники послуг та користувачі, які платять поставникам за надання послуг. Смарт-контракт здійснює валідацію даних, перевіряючи, чи пройшов платіж, розраховує комісію та суми, що призначені для відправки постачальникам. У ланцюжку буде тільки три учасники – постачальник, споживач та банк, – і ситуація, коли гроші, сплачені за газ і електрику, можуть бути присвоєні посередником, буде виключена.

Ще одним прикладом є управління ланцюжком поставок. Смарт-контракти дозволяють проконтролювати весь ланцюжок доставки товару: від відвантаження виробником до надходження до кінцевого споживача. Вся інформація про переміщення товару в незмінному вигляді зберігатиметься на блокчейні, а смарт-контракти контролюватимуть дотримання усіх умов поставки.

Нерухомість також є потенційною сферою для поширення смарт-контрактів. З їх допомогою, наприклад, можна регулювати взаємини забудовників, банків і майбутніх власників. Як тільки банк схвалює покупцеві квартири іпотечний кредит, смарт-контракт автоматично ініціює процес оформлення квартири у власність.

Можливе використання смарт-контрактів між рахунками материнських і дочірніх компаній. Наприклад, якщо на рахунку дочірнього підприємства є залишки, банк автоматично переводить їх на рахунок материнської компанії. Якщо, навпаки, дочірній компанії необхідно зробити платіж, а коштів на рахунку недостатньо, банк може провести платіж, зарезервувавши відповідну суму на рахунку материнської компанії. Завдяки смарт-контрактами подібні операції будуть здійснюватися набагато простіше і швидше.

Медицина займає значну частину економіки у більшості країн світу. Враховуючи кількість проведених транзакцій у медичній сфері, можна зробити висновок, про величезні об’єми даних, що передаються щодня. Ці дані можуть бути як загальнодоступними, так і конфіденційними, тому безпечне збереження даних – важлива задача медичних закладів. Блокчейни можуть використовуватися для зберігання різних даних про здоров'я таким чином, щоб вони були точними, повністю зашифрованими і підписувалися в цифровому вигляді. Це мінімізує ризик шахрайства і спрощує спілкування пацієнтів, страховиків і організацій охорони здоров'я один з одним та дозволяє створювати смарт-контракти, які миттєво передають достовірно точні дані про стан здоров'я пацієнтів.

Застосування смарт-контрактів також має місце у фінансовій сфері. Вони можуть запропонувати безліч корисних додатків для банків, що зможуть ефективно координувати юридичні контракти в форматі смарт-контрактів. Це допоможе усунути затримки, які зазвичай викликані централізованими інститутами, і дозволить смарт-контрактами ініціювати автоматичні виплати фіксованих валют з банківських рахунків після виконання умов контракту.

Аналогічним чином ці ж контракти можуть гарантувати виконання вимог податкових органів і автоматичну відправку відповідних звітів.

Смарт-контракти знаходяться на стадії впровадження у сучасне життя, і з кожним роком, сфер для їхнього застосування ставатиме дедалі більше.

2.7. Мова Solidity для платформи Ethereum

Існує ряд задач, де використання популярних високорівневих мов програмування неефективне, тому виникає потреба у розробці нових фреймворків та мов програмування. Саме для написання смарт-контрактів написано окрему мову програмування, що має назву Solidity. Solidity не отримала нічого нового, вона також містить змінні, функції, класи, арифметичні операції та інше. Solidity працює з «контрактом», який створюється з цих конструкцій.

Solidity - це нова високорівнева мова для віртуальної машини Ethereum з синтаксисом, схожим на JavaScript. Ця нова мова являє собою сукупність угод з мережевих технологій, мови асемблера і веброзробки. Програми транслюються в байткод EVM. Дозволяє розробникам створювати самодостатні додатки, що містять бізнес-логіку, результуючу у незмінні транзакційні записи блокчейна [23].

Створена в 2014 році командою програмістів під керівництвом Крістіана Райтвізнера на основі ідеї Гевіна Вуда. Схожість синтаксису на JavaScript – це розрахунок на швидку адаптацію розробників, які до цього на ньому розробляли подібні протоколи.

Не дивлячись на схожість з JS, Solidity містить декілька відмінностей:

— статично типізована мова програмування, динамічними є тільки типи, що повертаються;

— складні змінні-члени для контрактів, включаючи довільно ієрархічні відображення і структури;

— підтримка успадкування, включаючи множинне успадкування з лінеаризацією C3.

Повноцінної версії мови не вийшло, тому багато функціоналів поки в урізаному вигляді, а кількість помилок, що виникають у процесі реалізації, досить велика, однак, з поставленими завданнями Solidity справляється, а візуально це все той же ECMAScript (рис. 10).

```
contract GavCoin
{
  mapping(address=>uint) balances;
  uint constant totalCoins = 100000000000;

  /// Endows creator of contract with 1m GAV.
  function GavCoin(){
      balances[msg.sender] = totalCoins;
  }

  /// Send $((valueInmGAV / 1000).fixed(0,3)) GAV from the account of $(message.caller.address()), to an account accessible only by
$(to.address()).
  function send(address to, uint256 valueInmGAV) {
    if (balances[msg.sender] >= valueInmGAV) {
      balances[to] += valueInmGAV;
      balances[msg.sender] -= valueInmGAV;
    }
  }

  /// getter function for the balance
  function balance(address who) constant returns (uint256 balanceInmGAV) {
    balanceInmGAV = balances[who];
  }

};
```

Рисунок 10 – Приклад програми на мові Solidity

Solidity підтримує всі загальні типи даних, які підтримуються об'єктно-орієнтованими мовами програмування. Виділяють такі типи даних [24]:

— bool - логічне значення true / false;

— int8-int256 - цілочисельне значення зі знаком (з кроком розмірності в 8 біт);

— uint8-uint256- цілочисельне значення без знака (з кроком розмірності в 8 біт);

— address - стандартний адресний тип (20 байт) для адресації в блокчейні контрактів та рахунків;

— bytes1-bytes32 - масив байтів з вказаним розміром;

— byte - синонім bytes1;

— bytes — динамічний масив байтів;

— string — рядок;

— enum – перерахування;

— struct — структура даних.

Незважаючи на вузький профіль застосування, Solidity доступна для великого числа IDE і редакторів. Серед них: Remix, Emacs, IntelliJ IDEA, Microsoft Visual Studio, Vim Syntastic, Solium, Ethereum Studio, Sublime Text, Etheratom, Atom Solidity Linter. Можна використовувати Solidity в браузері без потреби завантажувати або компілювати що-небудь.

Solidity – це мова програмування, яка робить кодування на платформах блокчейна надзвичайно простим. Будучи простою для розуміння і відносно простою у використанні, Solidity має безліч застосувань. Найчастіше використовується у автоматизованому голосуванні, краудфандінгу, сліпих аукціонах, транзакціях з підтвердженням декількома підписами (мультипідписи) та у емісіях цифрових токенів.

3. АНАЛІЗ РОЗРОБКИ ТА РОБОТИ ПРОГРАМНОГО ПРОДУКТУ

3.1. Вибір та підготовка середовища для здійснення розробки смарт-контракту

Другий розділ дипломної роботи описує принцип роботи смарт-контрактів на основі платформи Blockchain Ethereum. Саме ця глобальна платформа дала можливість створювати смарт-контракти та застосовувати їх на практиці.

Вузол мережі Ethereum вирішено створити на базі клієнту Go Ethereum. Це програмне забезпечення, яке являє собою реалізацію протоколу Ethereum, реалізоване на мові програмування Go та доступне у вигляді програми Geth. На базі Geth можна створити багатофункціональний вузол мережі Ethereum [25].

Для роботи з вузлом Geth можна використовувати інтерфейс командного рядка, а також програмний інтерфейс. Використовуючи ці інтерфейси, а також різноманітні фреймворки можна створювати програми, шо працюють з вузлами Ethereum, практично на всіх сучасних мовах програмування.

Клієнт Geth може працювати на платформах, що містять підтримку мови Go, наприклад, Windows, Linux, iOS, Mac OSX, Raspberry Pi та інші.

Для розробки даного вузла мережі використовуватимесь Linux Mint, що заснований на операційній системі Ubuntu.

Далі необхідно перейти до установки Geth, для цього потрібно виконати наступні команди (рис. 11):

```
$ sudo apt-get install software-properties-common
$ sudo apt-get install build-essential
$ sudo add-apt-repository -y ppa:ethereum/ethereum
$ sudo apt-get update
$ sudo apt-get install ethereum
```

Рисунок 11 – Команди для установки Geth

Після установки, можна перевірити версію Geth та переходити до створення приватного блокчейну. Було установлено стабільну версію Geth 1.9.14 та Go версії 1.14.2 (рис. 12).

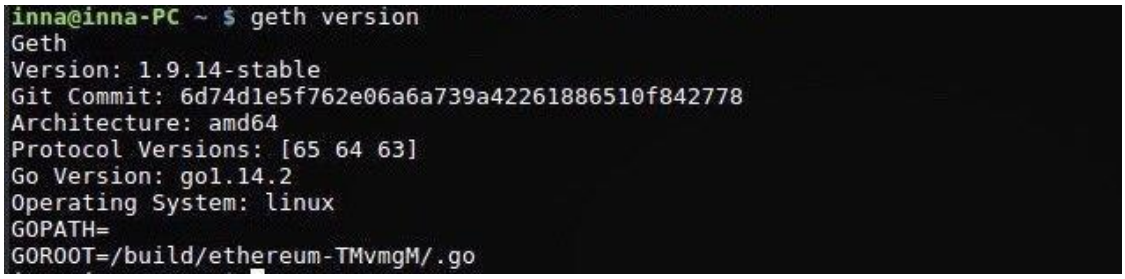

Рисунок 12 – Установлена версія Geth та Go

3.2. Створення приватного блокчейну на базі клієнту Go Ethereum

Для створення смарт-контракту необхідно використовувати блокчейн. Як було описано у попередніх розділах, він може бути приватним або ж публічним. У публічних блокчейнах первинний блок виникає разом із створенням мережі. Оскільки, створюється приватний блокчейн, то задля цього необхідно перш за все створити генезесний блок блокчейну, тому формуємо файл genesis.json (рис. 13).

```
{
"config": {
    "chainId": 1999,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0
},
"difficulty": "10",
"gasLimit": "5100000",
"alloc": {}
}
```

Рисунок 13 – Вміст файлу genesis.json

Блок config файлу містить змінні конфігурації мережі Ethereum, а поля виконують наступні функції:

- chainId – містить ідентифікатор блокчейну та використовується для захисту від копіювання, неавторизованих дій користувача. При створення власної приватної мережі, дане значення може бути будь-яким, окрім відомих значень, що відповідають публічним мережам;
- homesteadBlock – вказує на використання другого релізу мережі Ethereum, під назвою Homestead;
- eip150Block, eip155Block, eip158Block – мають значення 0, оскільки, при створення даного блокчейну хардфорк не буде здійснюватись;
- difficulty – впливає на час генерації нових блоків блокчейну;
- gasLimit – задає межу розходу газу в рамках блокчейну;
- alloc – дозволяє створювати гаманці та наповняти їх тестовими ефірами.

Маючи генезесний блок, можна перейти до створення приватного блокчейну. Створюємо акаунт операцією «$ geth –datadir node1 account new», де datadir вказує шлях до робочого каталогу, з підкаталогом node1, а account new – виведе в консоль так званий адрес вузла. При створенні акаунту також запрошується пароль, який є приватним паролем блокчейн акаунту, та повинен зберігатись у надійному, безпечному місці, оскільки, він не підлягає відновленню (рис. 14).

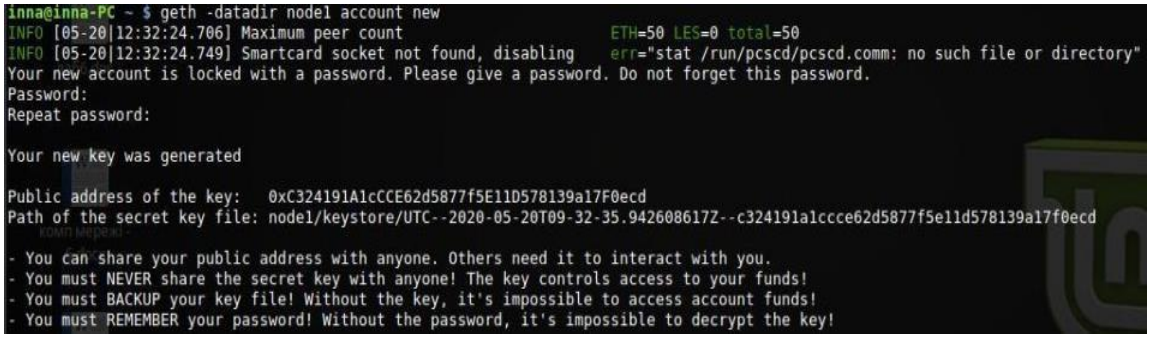

Рисунок 14 – Створення нового акаунту

Для ініціалізації акаунту з домашнього каталогу виконується команда «$ geth –datadir node1 init genesis.json». Команда виконається на виведе на консоль результат своєї роботи (рис. 21).

Рисунок 15 – Ініціалізація вузла з початковим блоком

Для роботи з вузлом та початку майнінгу використовуємо паралельно два вікна консолі. У першому для запуску вводимо команду «$ geth -etherbase '0xC324191A1cCCE62d5877f5E11D578139a17F0ecd' -datadir node1 -nodiscover -mine -minerthreads 1 -maxpeers 0 -verbosity 3 -networkid 98760 -rpc -rpcapi='db, eth, net, web3, personal, web3' console», де параметром команди etherbase є адреса вузла, отримана при початковому створенні акаунту. А також за допомогою параметру rpcapi, вказано перелік програмних інтерфейсів, які повинен надавати вузол. Параметр console запускає інтерактивну консоль JavaScript.

Команда виведе на екран повідомлення про хід ініціалізації, а після завершення процесу генерації, з другого вікна консолі необхідно підключитись до вузла, командою attach.

Надалі, користуватимемось інтерактивною консоллю та фреймоворком Web3. Web3.js є офіційною бібліотекою для роботи з блокчейном Ethereum, та дозволяє абстрагуватися від внутрішньої механіки Ethereum і працювати з мережею та смарт-контрактами так, ніби це звичайні javascript-об'єкти [26].

3.3. Розробка, опис та тестування смарт-контракту

3.3.1. Загальний опис

Смарт-контракт є автономним скриптом, написаним за допомогою мови Solidity, що зберігається у мережі блокчейн, а також визначає умови, з якими згодні всі сторони, що використовують контракт. Це описана структура даних, та набір методів, для роботи з ними. Перед публікацією контракту у мережі Euthereum, його необхідно компілювати у байт-код. Цей код зберігається у мережі, за допомогою транзакції. При публікації контракту, стає доступним його адреса, за допомогою якої децентралізовані додатки можуть використовувати методи контракту.

Розроблений смарт-контракт описує власну монету, а також містить функції для її переказу, та передбачає захищену процедуру доступу до неї. Він реалізує наступні функції:

— внесення початкових даних контракту;

— перевірка балансу та кількості монет;

— перевірка адреси власника контракту, та захист від його зміни;

— переказ коштів між учасниками, а також встановлення ліміту на зняття коштів;

— випуск нових монет на баланс будь-якого користувача

— відслідковування власником будь-якого пересилання монет

Опис власної монети та її використання може містить значно більше функцій, проте, описані функції є фундаментальними, а сам контракт може бути розширено шляхом комбінацій та додаванням нових функцій. Основним завданням даного контракту є реалізація функцій для надійного захисту коштів власника монети. Для забезпечення цілісної роботи смарт-контракту, він повинен містити функції та події, що повинні виникати при певних діях з монетою. Така монета може слугувати для ведення грошових переказів у державних установах, підприємствах, оскільки, є повністю захищеною та може контролюватись її власниками, за допомогою відповідних функцій.

3.3.2. Опис полів, функцій та подій смарт-контракту

Основними полями контракту є повна назва монети, коротка назва, що буде відображатись у гаманцях та на валютних біржах, а також поле цілечисельного типу, що міститиме кількість знаків після коми, для повноцінного функціонування монет (рис. 16).

```
string public constant name = "Coin Token";//повна назва монети
string public constant symbol = "CT";//скорочена назва монети
uint32 public constant decimals = 18;//кількість знаків після коми
uint public totalSupply = 0;//початковий баланс
```

Рисунок 16 – Початкові поля смарт-контракту

Також смарт-контракт повинен містити поле, що містить інформацію про адреси, що містять доступ до дій з монетами. Для цього використовуємо подвійний mapping, де перший ключ – це адреса на зняття з якої надається дозвіл, а другий ключ – користувач, який отримує дозвіл (рис. 17).

```
mapping (address => mapping (address => uint)) allowed;//поле дозволу
```

Рисунок 17 – Поле обмежень смарт-контракту

Баланси усіх користувачів зберігаються у функції balances, де кожній адресі користувача відповідає кількість монет (рис. 18). Функція transfer, вираховує з балансу користувача вказану кількість монет, проте, передбачає перевірку на доступну кількість монет для отримання та переповнення. Після, створює подію і ідентифікує успішний переказ монет (рис. 18).

```
function balanceOf(address _owner) public constant returns (uint balance) {
        return balances[_owner];//отримання балансу адреси
    }

function transfer(address _to, uint _value) returns (bool success) public {
    if(balances[msg.sender] >= _value && balances[_to] + _value >= balances[_to]) {//основні перевірки
        balances[msg.sender] -= _value; //зняття value монет у вказаної адреси
        balances[_to] += _value;//додавання value монет користувачу
        Transfer(msg.sender, _to, _value);//виклик події
        return true;//ідентифікатор успішного переказу
    }
    return false;
}
```

Рисунок 18 – Функції балансу та переказу коштів

Функція allowance повертає кількість монет, що дозволив знімати зі свого рахунку користувач, а функція approve дозволяє користувачеві знімати з рахунку кошти сумою не більше вказаного значення (рис. 19). На основі цього дозволу функціонуватиме transferFrom.

```
function allowance(address _owner, address _spender) public constant returns (uint remaining) {
    return allowed[_owner][_spender];//сума дозволена користувачеві _spender для зняття з рахунку _owner
}

function approve(address _spender, uint _value) public returns (bool success) {
    allowed[msg.sender][_spender] = _value;дозвіл на зняття _value монет користувачу _spender
    Approval(msg.sender, _spender, _value);//виклик події
    return true;
}
```

Рисунок 19 – Фунції надання дозволів

Функція transferFrom має такі ж перевірки, як і функція transfer, проте, вона міститиме перевірку на дозволену кількість отриманих монет. Користувач повинен мати дозвіл на переміщення монет між адресами, щоб будь-який бажаючий не зміг керувати чужими гаманцями. Фактично ця

функція дозволяє довіреній особі розпоряджатися певним обсягом монет на рахунку (рис. 20).

```
function transferFrom(address _from, address _to, uint _value) public returns (bool success) {
    if( allowed[_from][msg.sender] >= _value &&
        balances[_from] >= _value
        && balances[_to] + _value >= balances[_to]) {//основні перевірки
        allowed[_from][msg.sender] -= _value;//зменшення дозволеної кількості монет на величину зняття
        balances[_from] -= _value;
        balances[_to] += _value;
        Transfer(_from, _to, _value);
        return true;
    }
    return false;
}
```

Рисунок 20 – Функція переказу

Ще однією функцією контракту, є функція newToken, яка передбачає випуск нових монет на баланс будь-якого користувача, а також має обмеження, що дозволяють робити це лише власнику контракту, використовуючи для цього спеціальний модифікатор (рис. 21).

```
function newToken(address _to, uint _value) public onlyOwner {
    assert(totalSupply + _value >= totalSupply && balances[_to] + _value >= balances[_to]);
    balances[_to] += _value;//перерахунок балансу
    totalSupply += _value;//перерахунок загальної кількості монет
}
```

Рисунок 21 – Функція випуску нових монет на вказану адресу

Події дозволяють зручно використовувати засоби ведення журналу EVM. Коли вони викликаються, вони зберігають аргументи в журналі транзакцій - спеціальній структурі даних у блокчейні. Ці журнали пов'язані з адресою контракту і будуть включені в ланцюжок блоків і залишатимуться там до тих пір, поки блок доступний. Цим вони забезпечують абсолютний контроль над веденням контракту, Дані журналу і подій недоступні з контрактів (навіть з контракту, який їх створив) [27].

У рамках даного контракту було створено два типи подій, які повинні виникати при будь-якому переміщенні монет, а також при отриманні доступу

на зняття монет. Події створюються відповідними функціями та за рахунок функціоналу та можливостей мови Solidity заносяться до журналу (рис. 22).

```
event Transfer(address indexed _from, address indexed _to, uint _value);//подія для функцій  transfer, transferFrom

event Approval(address indexed _owner, address indexed _spender, uint _value);//подія для функції approve
```

Рисунок 22 – Події смарт-контракту

3.3.3. Тестування смарт-контракту

Перед публікацією смарт-контракту у приватну блокчейн мережу, можна провести його тестуваня у інтегрованому середовищі розробки Remix. Середовище розробки Remix надає великий набір допоміжних засобів для налагодження, статичного аналізу і публікації коду контракту. Написаний на JavaScript, Remix підтримує як використання в браузері, так і локально [28].

Компілюємо смарт-контракт у середовищі, та проводимо тестування існуючих функцій (рис. 23, 24).

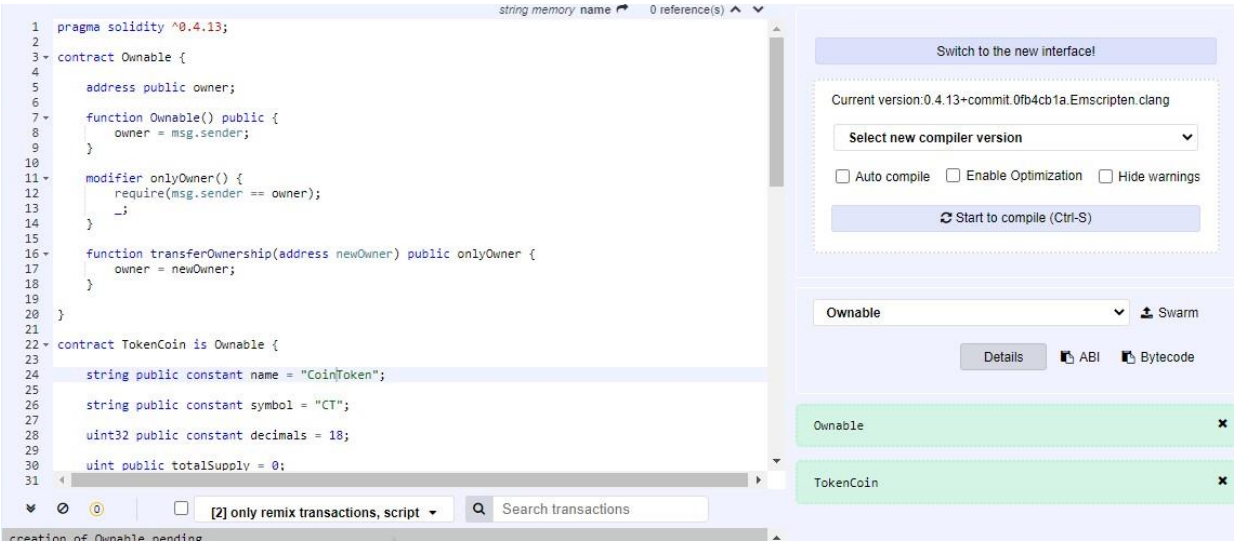

Рисунок 23 – Компіляція смарт-контракту в середовищі Remix

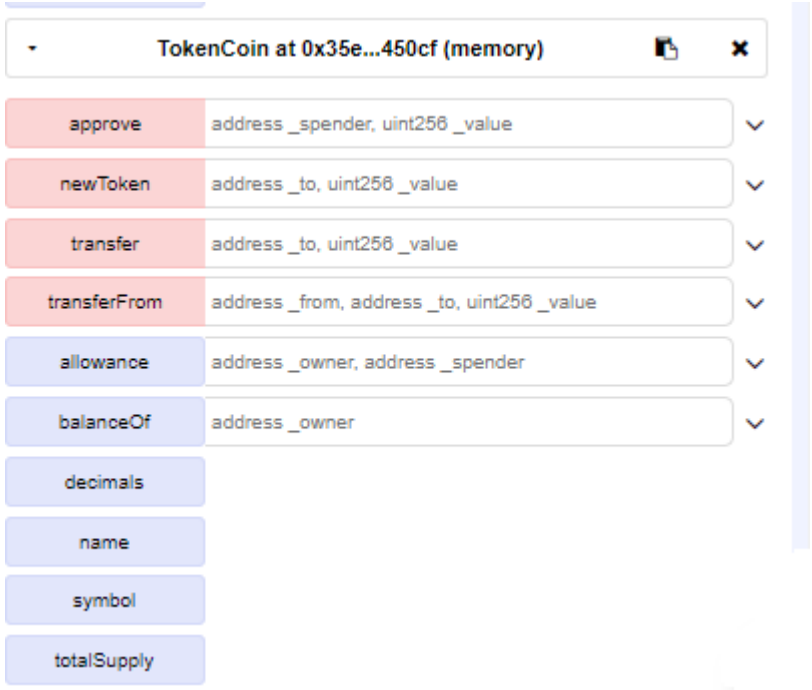

Рисунок 24 – Взаємодія з існуючими функціями та параметрами

Тестуємо функції переказу монет, спробуємо переказати монети на довільну адресу, враховуючи те, що на рахунку знаходиться 0 монет (рис. 25).

Рисунок 25 – Тестування функції transfer

Функція повернула false, тому що монет на нашому рахунку зараз немає, отже перевірка спрацювала. Те що монети не транспортувалися також можна переконатись викликавши функцію balance для адреси на який намагались перемістити монети.

Використавши функцію newToken, з адресою власника контракту, успішно випущено монети, що можна перевірити функцією balance (рис. 26, 27).

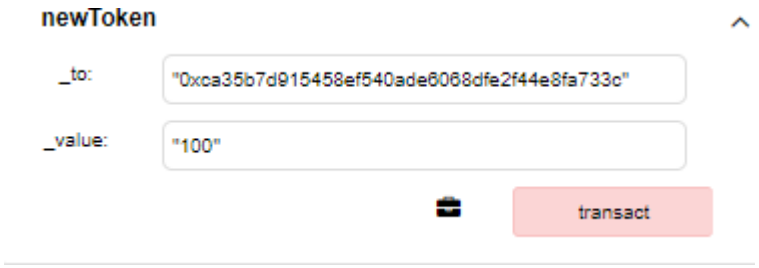

Рисунок 26 – Випуск нових монет

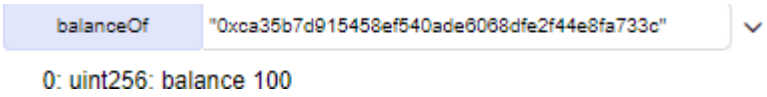

Рисунок 27 – Зміна стану рахунку

Маючи монети на рахунку, можна перевірити роботу функцій переказу. Для цього вказуємо адресу та кількість монет, та перевіряємо результат роботи функції (рис. 28).

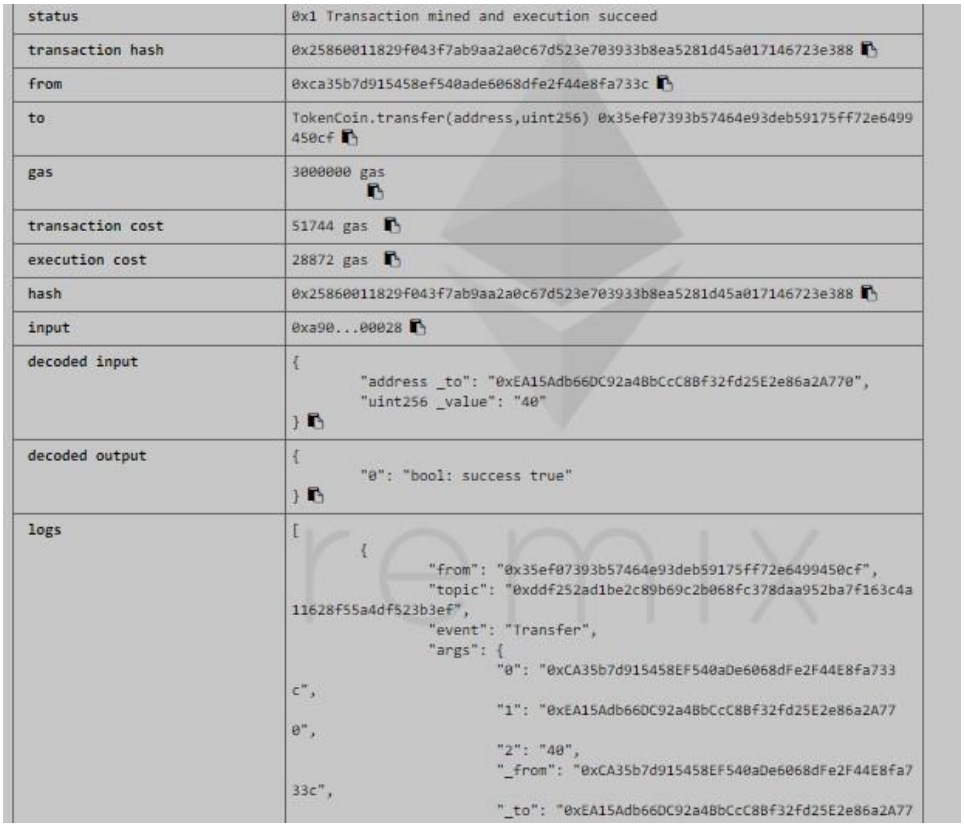

Рисунок 28 – Успішне виконання переказу

Перевіривши роботу функцій, можна перейти до публікації контракту у створену приватну блокчейн мережу.

3.4. Публікація смарт-контракту

При компіляції вихідного тексту контракту мовою Solidity, створюється не лише бінарний файл програми, але і так званий бінарний інтерфейс додатку Application Binary Interface (ABI), які необхідні для запуску контракту у приватній блокчейн мережі [28].

Інтерфейс ABI представляє собою специфікацію параметрів і значень методів контракту. Його та бінарний код програми можна також отримати за допомогою середовища Remix. Для цього необхідно скористатись блоком WEB3DEPLOY (рис. 29).

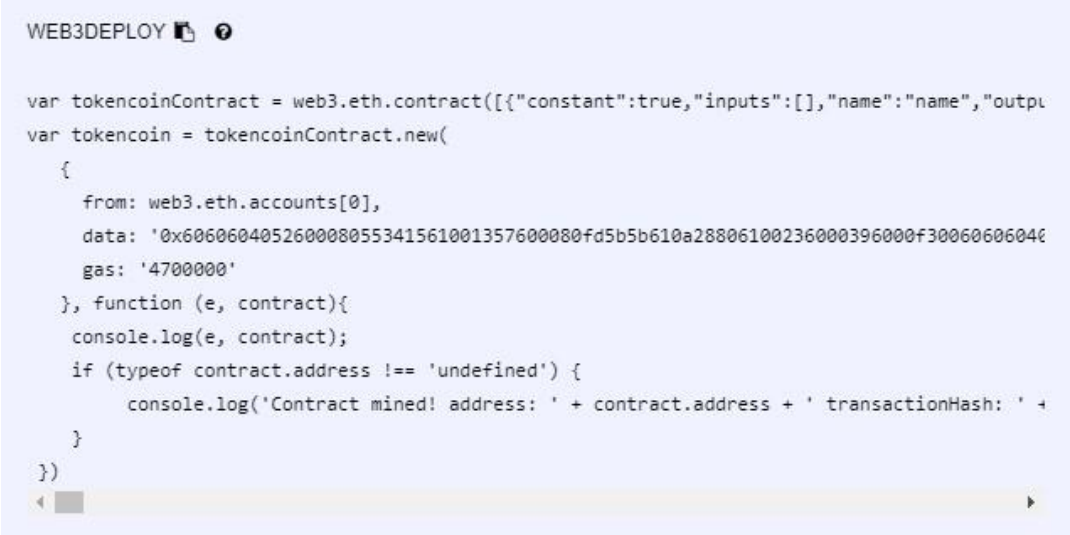

Рисунок 29 – Блок WEB3DEPLOY

Код, що містить даний блок представляє собою програму мовою JavaScript, яка викликає метод web3.eth.contract, призначений для створення об’єкту контракту. В якості параметрів методу передається масив ABI, у якому визначені функції контракту [29].

Для публікації контракту достатньо лише вставити код з блоку WEB3DEPLOY у консоль Web3 приватного блокчейну, та отримати результат успішної публікації контракту (рис. 30).

```
Contract mined! address:
0x1c766c3cd0114771c0e9e45e54f4924b3f81f0d8
transactionHash:
0xcf6af22b774045e7730cd8862d9bef36b1d05dac46c57a491eea
2a386377e97e
```

Рисунок 30 – Успішна публікація контракту

За допомогою функцій фреймворку Web3, можна перевірити стан транзакцій, а також провести взаємодію з контрактом. Протестуємо функцію видачі нових монет, у якості першого параметру, функції передається значення, а в якості другого – акаунт, від імені якого викликається функція.

Оскільки, дана функція доступна лише для власника контракту, вказуємо його адресу (рис. 31).

```
> CoinToken.newToken(777, {from:
"0x4f744742ac711fd111c7a983176db1d48d29f413"})
  "0x429eb0bbbc50cca55c8446adca8bdd5ed4c7df6fdc32930f7
e14ce7ccb51ea51"
```

Рисунок 31 – Виклик функції newToken

У вікні, де запущено вузол мережі, отримаємо повідомлення про запуск транзакції, яка викликала функцію newToken, хеш цієї транзакції та адресу контракту (рис. 32)

```
INFO [02-18|03:35:26.877] Submitted
transaction                    fullhash=0x429eb0bbbc50
cca55c8446adca8bdd5ed4c7df6fdc32930f7e14ce7ccb51ea51
recipient=0x1C766C3CD0114771C0e9E45e54F4924B3F81f0d8
```

Рисунок 32 – Повідомлення про запуск транзакції

Таким чином, за допомогою фреймворку Web3.js смарт-контракт опубліковано у створеній блокчейн мережі, та має всі можливості для взаємодії з функціями контракту.

ВИСНОВКИ

У процесі виконання дипломного проєкту, було проаналізовано проблему цілісності інформації безпеки, а на основі вивченого та опрацьованого матеріалу про загрози інформаційної безпеки, методи та підходи до їх рішень, а також про технологію блокчейн та смарт-контракти на основі блокчейну Ethereum, було розроблено смарт-контракт, що реалізовано у приватній блокчейн-мережі, доступ до якої є захищеним від несанкціонованого доступу, а функції контракту можливі лише тим користувачам, які мають на це встановлені права доступу.

У першому розділі проведено обґрунтування вибору даної теми, та аналіз існуючих проблем. Цілісність даних є однією з найважливіших проблем у наш час, а збитки, що несуть за собою загрози інформаційної безпеки є колосальними.

Дослідивши існуючі методи та загрози, зроблено висновок про те, що система інформаційної безпеки не є досконалою та має вразливі місця. Таким чином, запропоновано впровадження технології Blockchain, у цілях забезпечення цілісності даних та усунення можливих загроз. Є ряд питань, де технологія знайшла своє застосування, проте, ще жодна сфера повністю не змінила підхід до вирішення існуючих питань, шляхом впровадження децентралізованих баз даних. Існують проблеми, які виникають при впровадження технології Blockchain, проте, оскільки технологія є новою, вона постійно розвивається, а разом з тим, кількість проблем та недоліків стає дедалі меншою.

Другий розділ дипломного проєкту детально описує технологію. Було вивчено усі теоретичні особливості, досліджено будову блокчейну, його види, а також розглянуто всі переваги та недоліки.

Окрім того, було розглянуто смарт-контракти, що реалізуються за допомогою технології Blockchain. Проаналізувавши існуючі платформи, було

вирішено створити смарт-контракт на основі Blockchain Ethereum. Дана платформа разом з своєю появою дала змогу розвинути смарт-контракти та забезпечила їх інтеграцію у додатки та реальне життя.

У третьому розділі розроблено структуру приватної мережі Blockchain Ethereum, за допомогою використання програми Geth, а також розроблено смарт-контракт, який протестовано за допомогою інтерфейсу Remix, а після цього доданий у приватний блокчейн. Взаємодію з розробленим контрактом у приватній блокчейн мережі реалізує фреймворк Web3.js, проте, для цього можуть використовуватись і інші програмні засоби.

Показано, що створений смарт-контракт повністю передбачає захист даних, а доступ до змін надається лише його власнику. У подальшому такий смарт-контракт може бути використаний різними компаніями, для забезпечення надійних грошових обмінів та операцій, а також він може мати розширений функціонал, шляхом удосконалення та комбінації існуючих функцій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Making sense of bitcoin, cryptocurrency and blockchain – [Електронний ресурс]. – Режим доступу: https://www.pwc.com/us/en/industries/financial-services/fintech/bitcoin-blockchain-cryptocurrency.html.
2. How Do Ethereum Smart Contracts Work – [Електронний ресурс]. – Режим доступу: https://www.coindesk.com/learn/ethereum-101/ethereum-smart-contracts-work.
3. Киреенко, А. Е. Современные проблемы в области информационной безопасности: классические угрозы, методы и средства их предотвращения / А. Е. Киреенко. — [Електронний ресурс]. – Режим доступу: https://moluch.ru/archive/38/4365/.
4. Анин Б. Ю. Защита компьютерной информации / Б. Ю. Анин – Санкт-Петербург: БХВ-Санкт-Петербург, 2000. – 384 с.
5. Остапов С. Е. Технології захисту інформації: навчальний посібник / С. Е. Остапов, С. П. Євсеєв, О. Г. Король. – Харків: ХНЕУ, 2013. – 476 с.
6. Полевий В. І. Система забезпечення інформаційної безпеки України в контексті системності інформаційних загроз / В. І. Полевий // Науковий вісник НА СБ України. − 2006. − № 24. − С. 153–162.
7. Бутузов В.М. Протидія комп’ютерній злочинності в Україні (системно-структурний аналіз) : монографія – К. : КИТ, 2010. – 408 с.
8. Guardtime Announces BLT, New Blockchain Standard for Digital Identity. — [Електронний ресурс]. – Режим доступу: https://guardtime.com/blog/blt-new-blockchain-standard-for-digital-identity.
9. Blockchain Security Controls — [Електронний ресурс]. – Режим доступу: https://www.primechaintech.com/docs/blockchain_security_controls.pdf.
10. Зараменских Е., Артемьев И. Интернет вещей. Исследования и область применения. – М: Инфра-М, 2016. — 188 с.

11. Основные проблемы повседневного внедрения блокчейн технологии— [Електронний ресурс]. – Режим доступу: https://cryptor.net/bitkoin-dlya-chaynikov/osnovnye-problemy-povsednevnogo-vnedreniya-blokcheyn-tehnologii.
12. Blockchain Guide Documentation — [Електронний ресурс]. – Режим доступу: https://readthedocs.org/projects/blockchain-guide/downloads/pdf/latest/.
13. Hash Function — [Електронний ресурс]. – Режим доступу: https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm.
14. What Is a Block in the Blockchain? — [Електронний ресурс]. – Режим доступу: https://medium.com/datadriveninvestor/what-is-a-block-in-the-blockchain-c7a420270373.
15. Блокчейн изнутри. — [Електронний ресурс]. – Режим доступу: https://vas3k.ru/blog/blockchain/.
16. Три распространённых вида блокчейна — [Електронний ресурс]. – Режим доступу: https://bitnovosti.com/2018/04/25/3-popular-types-of-blockchains/.
17. Advantages and Disadvantages Of Blockchain Technology блокчейна — [Електронний ресурс]. – Режим доступу: https://data-flair.training/blogs/advantages-and-disadvantages-of-blockchain/.
18. Proof-of-Work vs. Proof-of-Stake — [Електронний ресурс]. – Режим доступу: https://medium.com/@hydrominer/proof-of-work-vs-proof-of-stake-7b3afe24f0cc.
19. Learn about Ethereum— [Електронний ресурс]. – Режим доступу: https://github.com/ethereumbook/ethereumbook/blob/develop/07smart-contracts-solidity.asciidoc#what-is-a-smart-contract.
20. What is Ethereum Gas? — [Електронний ресурс]. – Режим доступу: https://blockgeeks.com/guides/ethereum-gas/.

21. Pros and Cons of Smart Contracts— [Електронний ресурс]. – Режим доступу: https://atozmarkets.com/news/pros-and-cons-of-smart-contracts/.
22. Мультипідпис (Multisig) — [Електронний ресурс]. – Режим доступу: https://exbase.io/uk/wiki/multipidpis.
23. Ресурсы для разработчиков — [Електронний ресурс]. – Режим доступу: https://ethereum.org/ru/developers/.
24. Solidity — [Електронний ресурс]. – Режим доступу: https://solidity.readthedocs.io/en/v0.6.8/.
25. Go Ethereum — [Електронний ресурс]. – Режим доступу: https://geth.ethereum.org/.
26. Web3.js - Ethereum JavaScript API — [Електронний ресурс]. – Режим доступу: https://web3js.readthedocs.io/en/v1.2.8/.
27. Solidity - Events — [Електронний ресурс]. – Режим доступу: https://www.tutorialspoint.com/solidity/solidity_events.htm.
28. Remix documentation — [Електронний ресурс]. – Режим доступу: https://remix-ide.readthedocs.io/en/latest/.
29. Understanding the Application Binary Interface — [Електронний ресурс]. – Режим доступу: https://link.springer.com/chapter/10.1007/978-1-4612-3192-9_25.